

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

COMME EXIGENCE PARTIELLE
À L'OBTENTION DE
LA MAÎTRISE EN GÉNIE ÉLECTRIQUE
M. Ing.

PAR
BTISAM MOURID

FILTRE À DÉCIMATION PARALLELISÉ

MONTRÉAL, LE 16 OCTOBRE 2003

© droits réservés de Btisam Mourid

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE :

M. François Gagnon, directeur de mémoire
Département de génie électrique à l'École de technologie supérieure

M. Christian Gargour, président du jury
Département de génie électrique à l'École de technologie supérieure

M. Michel Forté, Spécialiste en conception numérique
Ultra Electronics

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 24 JUILLET 2003

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

FILTRE À DÉCIMATION PARALLELISÉ

Btisam Mourid

SOMMAIRE

Plusieurs applications radio utilisent de plus en plus de hautes fréquences d'échantillonnage qui permettent de numériser directement les signaux RF. Malheureusement, la vitesse de l'unité de traitement numérique des signaux actuellement disponible est limitée. Elle ne permet pas d'atteindre des hautes fréquences d'horloge.

L'objectif principal de ce mémoire est justement d'étudier et de développer une architecture prallelisée de filtres décimateurs qui offre la possibilité d'opérer à une haute fréquence d'échantillonnage de l'ordre de 1 GHz. Cette architecture doit être programmable avec les moyens technologiques disponibles tels que FPGA. Elle doit être également efficace avec une complexité acceptable. Ainsi, deux techniques ont été étudiées. La première technique consiste à utiliser des filtres numériques RII. La seconde technique consiste à l'utilisation des filtres Cascaded Integrate and Comb (CIC) où plusieurs structures ont été analysées et évaluées.

PARALLELIZED DECIMATING FILTERS

Btisam Mourid

ABSTRACT

Many software radio applications use high sampling frequencies in order to digitize signals at RF, which reduces the number of required tuneable RF components at the receiver. Due to various limitations of current FPGA technology, the sampled signal needs to be decimated efficiently with low complexity.

Thus, the goal of this project is to develop an appropriate architecture of programmable digital decimating filter. This parallelized architecture allows filtering signal sampled at 1 GHz with reasonable complexity. Hence, two techniques are considered. First one consists of using IIR filter. The second one uses the Cascaded Integrate and Comb filter (CIC), which a number of various architectures are evaluated and compared.

REMERCIEMENTS

J'aimerais remercier tout d'abord mon directeur de recherche Dr. François Gagnon pour ses précieux conseils. Son esprit scientifique d'une grande envergure m'a facilité la tâche de ma recherche.

J'aimerais remercier également Ultra electronics pour son support financier dans le cadre de la chaire de télécommunication sans fil dont le Dr. François Gagnon est titulaire.

J'aimerais remercier aussi les membres du jury messieurs C.Gargour et Michel Forté.

J'aimerais remercier tout particulièrement le professeur Marcel Gabrea pour ses remarques pertinentes.

Je remercie aussi tous mes collègues du laboratoire LACIME, en particulier Marc Lauzon et mon ami Hugo Létourneau pour la correction et la lecture de mes textes.

Je tiens à remercier tous mes amis (es) pour leur soutien moral dans les moments difficiles, particulièrement Isabelle, Solange, Céline, Nathalie, Fayçal , Raouf et Nabil.

J'aimerais remercier chaleureusement mes chers parents, mes sœurs et frères qui n'ont cessé de m'encourager dans mon cheminement et ce même de l'autre côté de l'atlantique.

Je remercie infiniment Luc pour son soutien moral et surtout sa patience durant toute la période de ce travail.

Finalement, merci à toutes les personnes qui ont contribué de près ou de loin à mener à terme ce travail.

TABLE DES MATIÈRES

	Page
SOMMAIRE	i
ABSTRACT	ii
REMERCIEMENTS	iii
TABLE DES MATIÈRES.....	v
LISTE DES FIGURES	viii
INTRODUCTION.....	1
CHAPITRE 1 CONCEPTS DE BASE	4
1.1 Signaux et informations.....	4
1.2 Classification des signaux	4
1.3 Traitement des signaux.....	6
1.4 Numérisation des signaux	6
1.5 Échantillonnage	7
1.5.1 Théorème d'échantillonnage	7
1.6 La quantification.....	9
1.6.1 Représentation numérique	9
1.6.2 Erreur de la quantification	11
1.7 Filtre et filtrage numériques	13
1.7.1 Transformée en Z	14
1.7.2 Structure de réalisation	16
1.7.2.1 Structure directe.....	16
1.7.2.2 Structure en cascade	16
1.7.2.3 Structure parallèle.....	17
1.7.2.4 Structure polyphase	17
1.8 Filtres multicaudence	18
1.8.1 Décimation	19
1.8.2 Interpolation	21

1.9	Conclusion.....	23
CHAPITRE 2	FILTRES NUMÉRIQUES À RÉPONSE IMPULSIONNELLE INFINIE.....	24
2.1	Présentation générale des filtres RII.....	24
2.1.1	Introduction	24
2.1.2	Expressions générales.....	25
2.2	Méthode de calcul des coefficients des filtres RII.....	26
2.2.1	Filtres de Butterworth.....	27
2.2.2	Filtres elliptiques	29
2.3	Conception d'un filtre RII passe- bas.....	30
2.3.1	Détermination des coefficients du filtre	30
2.3.2	Réponse fréquentielle	32
2.3.2.1	Module.....	32
2.3.2.2	Déphasage.....	33
2.3.3	Concept de stabilité	34
2.4	Structures de réalisation des filtres RII	37
2.4.1	Structures directes	37
2.4.2	Limitation du nombre de bits des coefficients	39
2.5	Autres techniques	41
2.6	Conclusion.....	44
CHAPITRE 3	FILTRES CIC DÉCIMATEURS.....	46
3.1	Introduction	46
3.2	Description générale.....	46
3.2.1	Bloc intégrateur	47
3.2.2	Bloc différentiateur.....	49
3.2.3	Filtre CIC complet.....	50
3.3	Théorie de conception des filtres CIC décimateurs.....	51
3.3.1	Paramètres de conception	51
3.3.1.1	Ordre du filtre	51

3.3.1.2	Largeur des registres.....	51
3.3.2	Réponse fréquentielle	54
3.4	Réalisation des filtres CIC décimateurs	56
3.4.1	Structure pipeline	56
3.4.2	Structure parallèle.....	57
3.4.3	Structure parallélisée améliorée	58
3.4.4	Calcul de la complexité des deux structures	59
3.4.5	Structure parallèle utilisant la décomposition polyphase	61
3.5	Conclusion.....	72
CHAPITRE 4	CONCEPTION DÉTAILLÉE DES FILTRES CIC DÉCIMATEURS...	74
4.1	Application de l'architecture proposée.....	74
4.2	Interprétation des résultats de simulation	77
4.3	Calcul de la complexité	81
4.4	Conclusion.....	82
CONCLUSION GÉNÉRALE		83
BIBLIOGRAPHIE.....		85

LISTE DES FIGURES

Figure 1	Signal numérique	5
Figure 2	Exemples de reconstitution de signal [9]	9
Figure 3	Représentation par signe et par module	10
Figure 4	Représentation par complément à 2	11
Figure 5	Quantification par arrondi.....	12
Figure 6	Quantification par troncature	13
Figure 7	Structure cascade.....	16
Figure 8	Structure parallèle	17
Figure 9	Structure polyphase.....	18
Figure 10	Représentation d'un décimateur	19
Figure 11	Exemple de décimation.....	20
Figure 12	Filtre de décimation	20
Figure 13	Module d'un simple filtre de décimation	21
Figure 14	Représentation d'un interpolateur	21
Figure 15	Exemple d'interpolation	22
Figure 16	Filtre d'interpolation.....	22
Figure 17	Module de la fonction de transfert $H(z)$ (2.19)	32
Figure 18	Phase de la fonction de transfert $H(z)$ (2.19)	33
Figure 19	Stabilité de la fonction de transfert $H(z)$ (2.19)	35
Figure 20	Comparaison du module (module1) avec tous les chiffres significatifs des coefficients et le module à quatre chiffres significatifs(module2).....	36
Figure 21	Structure directe de type I pour les filtres numériques RII	38
Figure 22	Structure directe de type II pour les filtres numériques RII.....	39
Figure 23	Schéma représentant le filtre RII en structure pipeline.....	42
Figure 24	Schéma représentant le filtre RII en combinaison linéaire	43
Figure 25	Structure de base d'un filtre CIC	47
Figure 26	Circuit équivalent à la structure de base d'un filtre CIC	47

Figure 27	Schéma d'un simple intégrateur.....	48
Figure 28	Schéma d'un simple différentiateur	49
Figure 29	Module normalisé de la réponse en fréquence du filtre CIC décimateur 3.2.2	55
Figure 30	Structure pipeline d'un CIC décimateur à deux étages.....	56
Figure 31	Structure parallèle de l'intégrateur du CIC.....	58
Figure 32	Structure parallélisée améliorée avec 8 entrées	59
Figure 33	Structure polyphase du filtre FIR d'ordre 6 avec 2 branches	62
Figure 34	Structure polyphase du filtre FIR d'ordre 6	63
Figure 35	Structure polyphase du filtre FIR d'ordre 6	63
Figure 36	Décimation multiétage utilisant un filtre CIC [19]	64
Figure 37	Module de la réponse fréquentielle du filtre CIC ($N=5$, $R=250$).....	67
Figure 38	Module de la réponse fréquentielle du filtre CIC	68
Figure 39	Module de la réponse fréquentielle du filtre CIC	69
Figure 40	Module de la réponse fréquentielle du premier étage	70
Figure 41	Module de la réponse fréquentielle du second étage	71
Figure 42	Module de la réponse fréquentielle du filtre au complet	72
Figure 43	Structure parallèle d'un CIC utilisant la décomposition polyphase.....	74
Figure 44	Exemple de réalisation pour $E_0(z)$	76
Figure 45	Diagramme bloc	77
Figure 46	Module de la réponse fréquentielle.....	79
Figure 47	Comparaison du module simulé avec le module théorique	80

INTRODUCTION

Actuellement, le développement spectaculaire de la technologie des circuits intégrés de plus en plus complexes et de plus en plus puissants permet une expansion importante des applications en télécommunication sans fil. Notamment, la radio mobile, la vidéo conférence numérique... Ces nouvelles applications nécessitent généralement de transmettre une quantité importante d'information à haute vitesse. De plus, l'utilisation de hautes fréquences d'échantillonnage permet de numériser les signaux directement en RF. L'utilisation de telles techniques permet de remplacer plusieurs circuits analogiques coûteux et peu flexibles. Cependant, le traitement numérique de ces signaux s'avère complexe. Car les dispositifs disponibles de traitement numérique des signaux, essentiellement les filtres numériques ne peuvent assurer une haute fréquence d'échantillonnage de l'ordre des gigahertz (GHz). Un autre défi pour les communications sans fil, c'est la réduction de la puissance consommée. Une des solutions est de réduire la tension d'alimentation [1]. Cette solution engendre l'augmentation des délais. Ce qui limite inévitablement la fréquence d'opération. Une solution efficace à ce problème est l'utilisation d'un traitement parallèle.

Effectivement, la technique de filtrage multiscalaire [2] a été introduite dans le but de réduire la vitesse de calcul dans les filtres numériques et ainsi augmenter la rapidité du traitement des signaux. Cette technique inclut les opérations de décimation et d'interpolation qui permettent de changer le facteur d'échantillonnage. L'introduction du filtrage multiscalaire a suscité l'intérêt de plusieurs chercheurs, Parmi lesquels, R.E.Crochiere et L.R.Rabiner qui ont été les premiers à présenter une conception optimale en terme de minimisation de nombre de multiplicateurs dans les filtres numériques à Réponse Impulsionnelle Finie (RIF) décimateurs multi-étage [3][4]. Ils ont apporté des contributions au plus haut niveau, dans le domaine de filtrage numérique, en particulier les filtres multiscalaire. Cette approche [3][4] qui consiste à réduire le nombre de multiplicateurs dans les filtres FIR a inspiré Peled et Liu [5] à

remplacer les multiplicateurs par des additionneurs. Puisque, la multiplication est un facteur représentatif de la complexité des filtres numériques. Dans la même optique, Goodman et Carry ont démontré qu'un choix judicieux et adéquat des coefficients des filtres demi bande décimateurs et interpolateurs peut être un moyen efficace pour la réalisation des filtres performants [6].

En 1981, un article de E.B Hogenauer [7] a introduit de nouveaux filtres. Cet article présente une méthode efficace et économique pour la conception des filtres muticadence décimateurs et interpolateurs. Cette méthode consiste à mettre en cascade deux blocs. Le premier bloc est constitué d'étages d'intégrateurs. Le deuxième bloc est constitué d'étages de différentiateurs. Ces filtres sont appelés Cascaded Integrate and Comb (CIC). Les filtres CIC sont plus performants que les filtres décrits par Peled et Liu [5]. Car ces derniers ont un facteur de décimation ou d'interpolation de 2 et l'atténuation dans les bandes images est très limitée. Le grand avantage des filtres CIC réside dans le fait que la réalisation de ces filtres ne nécessite ni multiplicateur ni stockage des coefficients des filtres, ce qui présente un excellent moyen soit d'augmenter la vitesse de traitement ou encore diminuer la taille des filtres. Toutefois, l'architecture telle que présentée n'est pas appropriée pour des applications à hautes fréquences.

L'objectif principal de notre recherche est d'étudier et de développer la conception d'une architecture de filtres décimateurs capables de filtrer un signal échantillonné à 1GHz sur des puces ne supportant que des cadences de 100 MHz. Évidemment, le filtre doit être programmable avec les moyens technologiques actuels. Pour ce faire, nous avons commencé par étudier toutes les possibilités qui permettent d'augmenter la vitesse de traitement. En premier lieu, l'architecture doit être parallèle. Puisque c'est un moyen efficace qui permet à un processus non seulement d'avoir une plus grande vitesse mais aussi de réduire la puissance consommée. En outre, l'utilisation optimisée de certains outils de conception des filtres numériques s'avère nécessaire. La première approche concerne le problème de l'approximation optimale pour le calcul des

coefficients des filtres numériques à Réponse Impulsionnelle Infinie (RII) satisfaisant aux spécifications techniques imposées (chapitre 2). Ensuite vient le choix de la structure du filtre avec toutes ses implications en ce qui concerne les performances (chapitre 3).

Ainsi, ce mémoire est composé de 4 chapitres : le premier chapitre introduit d'une façon brève mais concise les outils indispensables au traitement numérique des signaux, et dans les chapitres suivants, sont développés les méthodes pour réaliser un filtre à décimation parallélisée. Le second chapitre développe la théorie des filtres numériques RII. Le troisième chapitre est consacré à une étude approfondie des filtres CIC. Cette étude inclut les détails de l'architecture que nous avons développée. Le quatrième chapitre présente la simulation de l'architecture proposée dans le chapitre précédent, ainsi que les résultats obtenus et la conclusion générale.

CHAPITRE 1

CONCEPTS DE BASE

Le but de ce premier chapitre n'est pas d'être une étude mathématique du signal, mais seulement un rappel succinct des notions fondamentales des outils de base du traitement numérique des signaux.

Ce premier chapitre sera centré sur les éléments de base de représentation des signaux en relation avec le filtrage numérique et plus particulièrement le filtrage multiscadence. Ce choix a été guidé par le souci de se conformer à faciliter l'analyse des méthodes de filtrage numérique utilisées dans les prochains chapitres.

1.1 Signaux et informations

D'une façon générale, un signal est défini comme toute manifestation sous forme physiquement observable et mesurable d'un phénomène. Par exemple, les signaux visuels sont des ondes de lumière apportant une information à nos yeux. Les signaux sonores sont des fluctuations de la pression de l'air transportant un message à nos oreilles. Dans un sens plus limité, un signal est la représentation physique de l'information qu'une source émet vers un récepteur. Dans un sens mathématique, le signal est représenté par une fonction réelle ou complexe à une ou plusieurs dimensions d'une ou plusieurs variables réelles ou entières.

1.2 Classification des signaux

Cette classification est basée sur la nature de l'évolution du signal. Ainsi, il existe plusieurs types de signaux dont les plus utilisés en traitement des signaux sont les suivants:

- Signal analogique: le signal analogique est un signal dont les variables indépendantes de la représentation mathématique, sont continues. Souvent, ces signaux sont modélisés par des fonctions de la variable t . Le support temporel du signal $x(t)$ peut être borné dans le temps.
- Signal discret: un signal discret appelé aussi signal échantillonné est un signal dont les variables indépendantes de la représentation mathématique sont discrètes.
- Signal numérique: c'est un signal discret, dont l'amplitude est aussi discrète. Un exemple de signal numérique est représenté sur la figure 1

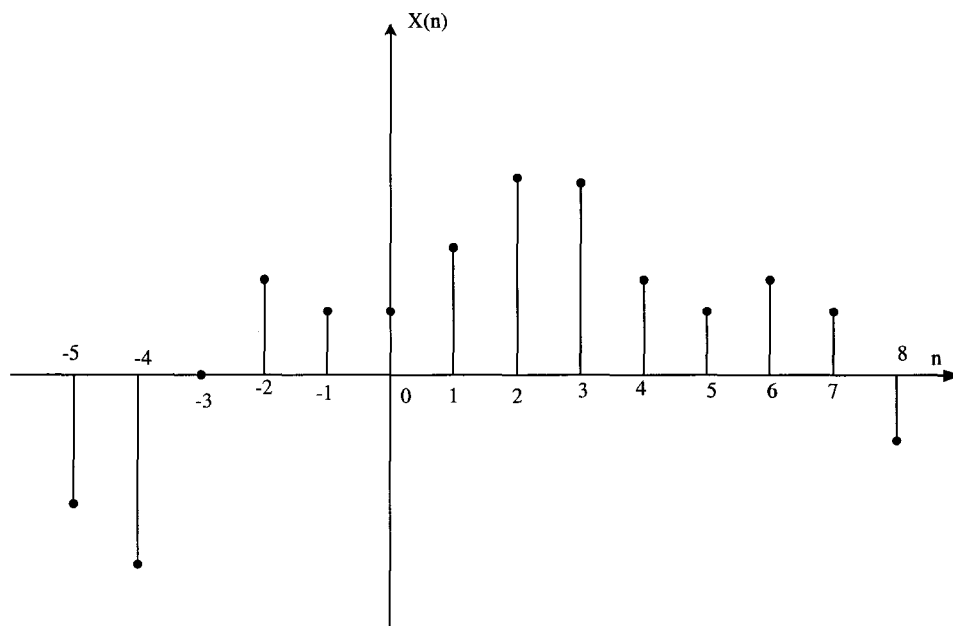


Figure 1 Signal numérique

1.3 Traitement des signaux

La théorie du signal ainsi que son traitement sont étroitement liés : la théorie du signal modélise et décrit les signaux, tandis que le traitement des signaux interprète et élabore d'autres signaux à l'aide de dispositifs ou systèmes de traitement des signaux d'entrée. Un système de traitement de signaux agit sur un signal d'entrée et produit à sa sortie un autre signal sous une forme plus appropriée pour satisfaire certaines exigences techniques requises par une application donnée. Les systèmes de traitement peuvent être classés de la même façon que les signaux. Les systèmes échantillonnés opèrent sur des signaux échantillonnés et produisent des signaux échantillonnés. Les systèmes numériques opèrent sur des signaux numériques et produisent des signaux numériques. Le traitement numérique d'un signal implique une représentation appropriée, soit une représentation en virgule fixe ou en virgule flottante.

1.4 Numérisation des signaux

La plupart des signaux sont de type analogique. Si on veut transmettre des informations sous forme numérique ou si on désire faire subir à celles-ci des transformations à l'aide des systèmes numériques, il faut d'abord convertir ces signaux en une suite de nombres codés sous forme binaire. Cette représentation est effectuée par une conversion analogique à numérique.

La conversion d'un signal analogique à un signal numérique s'appuie sur une double approximation. La première approximation s'effectue dans l'espace des temps: le signal $x(t)$ est remplacé par ses valeurs $x(nT)$ à des instants multiples entiers d'une durée T ; c'est ce qu'on appelle échantillonnage. La deuxième approximation s'effectue au niveau des amplitudes. Chaque valeur $x(nT)$ est approchée par un multiple entier d'un pas élémentaire noté q ; c'est ce qu'on appelle quantification.

1.5 Échantillonnage

L'échantillonnage est défini comme le prélèvement régulier de valeurs $x(nT_e)$ à des instants multiples entiers d'une durée T_e . Cette durée représente la période d'échantillonnage d'un signal analogique fonction du temps $x(t)$.

Quand un signal analogique quelconque $x(t)$ est échantillonné, il est essentiel de savoir s'il est possible de le reconstituer au besoin, à partir des échantillons $x(nT_e)$.

L'intérêt principal du théorème d'échantillonnage est d'établir la condition que doit satisfaire la fréquence d'échantillonnage f_e pour avoir la possibilité de reconstitution du signal d'origine sans perte d'information.

1.5.1 Théorème d'échantillonnage

Ce théorème énonce les conditions que doit respecter la suite des échantillons $x(nT_e)$ d'un signal pour représenter correctement ce signal. Un signal est considéré correctement représenté par les échantillons $x(nT_e)$ s'il est possible de reconstruire le signal d'origine $x(t)$ à partir de ses échantillons $x(nT_e)$ prélevés avec la période T_e . Pour reconstruire le signal d'origine $x(t)$, il faut éliminer la périodicité du spectre de fréquence qui a été introduite par l'échantillonnage.

L'élimination de cette périodicité ou les bandes images, peut être réalisée par un filtre passe-bas dont la fonction de transfert $H(f)$ vaut $1/f_e$ jusqu'à la fréquence $f_e/2$ et zéro pour les fréquences supérieures. À la sortie de ce filtre on trouve le signal continu $x(t)$. Ce signal peut être exprimé en fonction des valeurs $x(nT)$ et a pour expression [8]:

$$\begin{aligned}
x(t) &= \int_{-\infty}^{+\infty} \left[\sum_{n=-\infty}^{+\infty} x(nT) \delta(t - nT) \right] \frac{\sin \pi(t - \theta)/T}{\pi(t - \theta)/T} d\theta \\
&= \sum_{n=-\infty}^{+\infty} x(nT) * \frac{\sin \pi(t/T - n)}{\pi(t/T - n)}
\end{aligned} \tag{1.1}$$

Quant à la réponse impulsionnelle du filtre $h(t)$, elle est sous la forme suivante [8] :

$$h(t) = \frac{\sin \pi t/T}{\pi t/T} \tag{1.2}$$

L'expression (1.1) montre que le signal de sortie $x(t)$ est en effet le produit de convolution de $x(nT)$ par la réponse impulsionnelle $h(t)$ (1.2).

Pour que la répétition périodique du spectre $S(f)$ ne déforme pas le motif répété, il faut que la fréquence de répétition f_e soit supérieure ou égale à deux fois la fréquence maximale. C'est le théorème de Nyquist. Dans le cas où cette condition ne serait pas respectée, un chevauchement de bandes images apparaît. Le filtre ne fournit pas un signal identique à celui d'origine.

Il est à noter que la fréquence d'échantillonnage d'un signal est déterminée par la largeur de bande qu'occupe ce signal. La figure 2 [9] montre quelques exemples de reconstitution de signal $x(t)$ après l'échantillonnage.

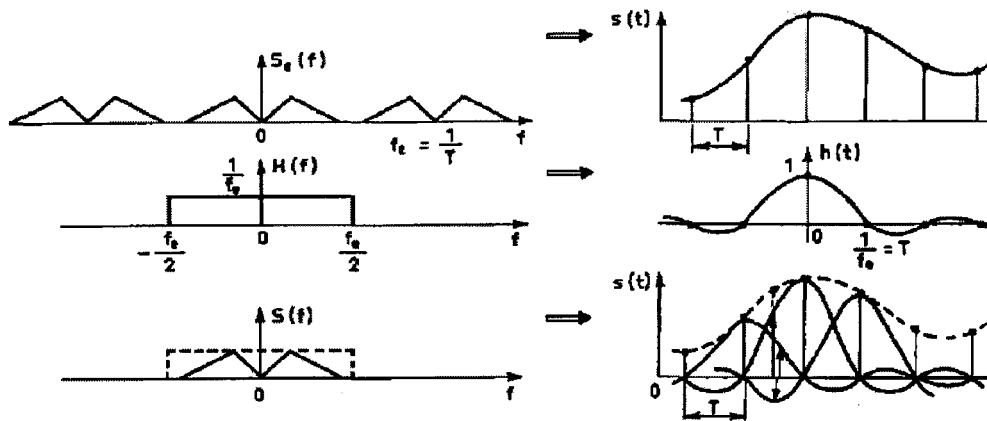


Figure 2 Exemples de reconstitution de signal [9]

1.6 La quantification

Le traitement numérique des signaux nécessite une représentation d'un nombre fini de valeurs discrètes, il faut donc quantifier le signal. Quantifier un signal $x(t)$ revient à une approximation de chaque valeur de ce signal par un multiple entier d'une quantité élémentaire, notée q et appelée pas de quantification. Lorsque q est constant et indépendant de l'amplitude, on parle d'une quantification uniforme; c'est le cas de la représentation des nombres en virgule fixe. Par contre, quand q est approximativement proportionnel à l'amplitude du signal, la représentation numérique se fait en virgule flottante. Un aperçu succinct de la représentation en virgule fixe sera donné au sous paragraphe 1.6.1.

1.6.1 Représentation numérique

En pratique, deux types de représentations sont souvent utilisés : La représentation en virgule fixe et la représentation en virgule flottante. La représentation en virgule fixe est une suite de valeurs représentables équidistantes sur l'axe réel. Souvent on choisit

l'ensemble symétrique par rapport à l'origine. Certainement, le nombre de valeurs représentables définit le nombre de bits nécessaire pour leur représentation.

Par exemple si la dynamique des signaux numériques est l'intervalle $[-1, +1]$ l'ensemble des valeurs représentables par bit est : $\{-1, \dots, -2.2^{-(b-1)}, -2^{-(b-1)}, 0, 2.2^{-(b-1)} \dots +1\}$

Une méthode simple de la représentation consiste à réserver un bit pour le signe et b-1 bits qui restent pour le module, c'est ce qu'on appelle, la représentation par signe et par module.

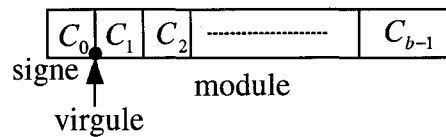


Figure 3 Représentation par signe et par module

La distance q entre deux valeurs successives, le pas de quantification est lié au nombre de bits de la représentation par signe et module par la relation suivante [10]:

$$q = \frac{x_{\max} - x_{\min}}{2^b - 2} \quad (1.3)$$

où x_{\max} représente la valeur la plus grande dans la dynamique d'un signal, x_{\min} la valeur la plus petite. Une autre méthode fréquemment utilisée est la représentation par complément à deux. Elle présente des avantages pour les opérations arithmétiques. Cette méthode consiste à remplacer le nombre négatif $-r$ par $2-r$ et à développer ce dernier comme fraction binaire

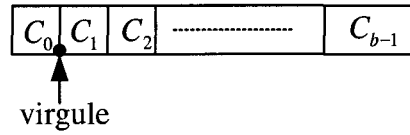


Figure 4 Représentation par complément à 2

La relation liant le pas de quantification et le nombre de bits est [10]:

$$q = \frac{x_{\max} - x_{\min}}{2^b - 1} \quad (1.4)$$

1.6.2 Erreur de la quantification

Comme la quantification est une approximation, il est évident qu'elle entraîne une erreur. L'erreur due à la quantification du signal $x(n)$ est une suite désignée par $e(n)$:

$$e(n) = x_q(n) - x(n) \quad (1.5)$$

La manière dont l'approximation est faite définit deux types de quantification. Le premier consiste à choisir la valeur représentable la plus proche de la valeur donnée, autrement dit, pour toute valeur du signal comprise entre $(n-1/2)q$ et $(n+1/2)q$ est arrondie à nq , d'où l'appellation de quantification par arrondi. C'est l'approximation qui conduit à une erreur de quantification minimale. La figure 5 représente un exemple de la quantification par arrondi, en représentation en virgule fixe. En général, l'erreur d'arrondi est bornée [11]:

$$e(n) = |x(n) - x_q(n)| \leq \frac{q}{2} \quad (1.6)$$

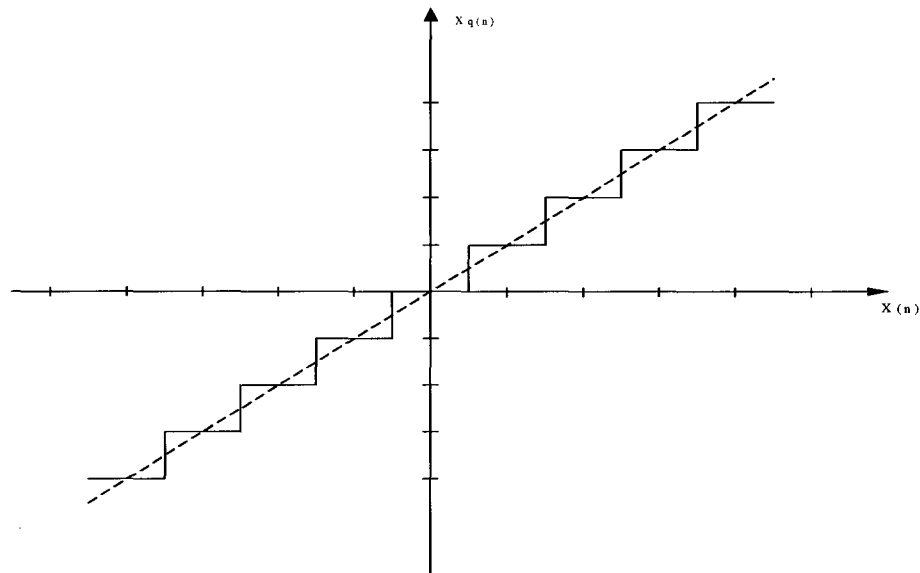


Figure 5 Quantification par arrondi

Le second type est une approximation par défaut. Elle consiste à choisir la valeur représentable la plus proche dont le module est inférieur à celui de la valeur donnée. Cette méthode est appelée quantification par troncature. La figure 6 montre un exemple de quantification par troncature avec le même pas de quantification q que l'exemple de la figure 5. La borne générale pour l'erreur est :

$$|e(n)| = q \quad (1.7)$$

Comparativement à l'erreur d'arrondi, l'erreur de troncature peut devenir deux fois plus grande. La troncature du module est utilisée lorsqu'on veut réduire le nombre de bit d'un signal à l'intérieur d'un filtre, quand le signal est représenté par le signe et le module. Il suffit d'éliminer les bits de poids les plus faibles qui sont en trop.

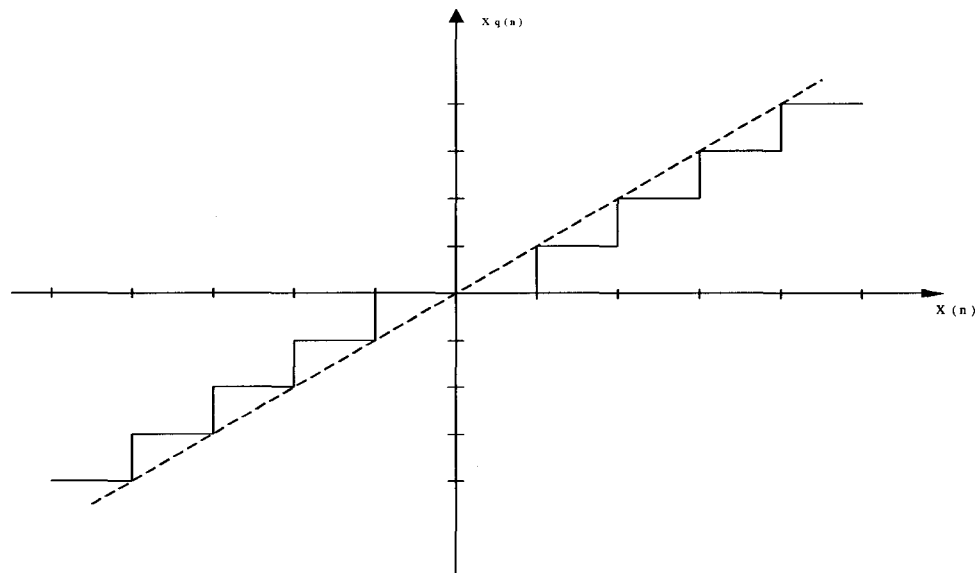


Figure 6 Quantification par troncature

1.7 Filtre et filtrage numériques

Un filtre numérique est un système à temps discret dans lequel toutes les variables (entrée, sortie....) sont sous forme numérique. Les coefficients du filtre sont également numériques. Toutes ces grandeurs sont quantifiées. Un filtre linéaire est caractérisé par une réponse impulsionnelle $h(n)$ permettant de calculer le signal de sortie $y(n)$ à partir du signal d'entrée $x(n)$ par le produit de convolution [12]

$$y(n) = h(n) * x(n) = \sum_i h(i).x(n-i) \quad (1.8)$$

Dans un filtre numérique de fréquence, la réponse impulsionnelle $h(n)$ ne dépend pas du signal traité. Dans le cas où cette réponse serait de durée finie, le filtre est dit filtre à Réponse Impulsionnelle Finie (RIF) ou filtre non récursif. Si la réponse est de durée infinie, le filtre est dit à Réponse Impulsionnelle Infinie (RII) ou filtre récursif.

La fonction de transfert $H(z)$, transformée en z de la réponse impulsionnelle $h(n)$ est une caractéristique importante couramment utilisée pour caractériser les filtres numériques. Dans le domaine z l'équation 1.8 s'écrit :

$$Y(z) = H(z).X(z) \quad (1.9)$$

où $X(z)$ représente la transformée en z de $x(n)$ et $Y(z)$ est la transformée en z de $y(n)$.

1.7.1 Transformée en Z

La transformée en z d'un signal $x(n)$ est définie comme suit [12]:

$$x(z) = \sum_{n=-\infty}^{+\infty} x(n)z^{-n} \quad (1.10)$$

où z est une variable complexe et $X(z)$ est une fonction complexe de variable z . L'ensemble des valeurs de z pour lesquelles la transformée en z existe, est la région de convergence de $X(z)$. La transformée de z possède des propriétés semblables à celles de la transformée de Fourier. Toutefois, les implications diffèrent suivant les cas. L'une des propriétés fondamentales de la transformée en z est la linéarité. La transformée en z d'une combinaison linéaire des signaux est la même combinaison linéaire des transformées en z correspondantes. Une autre propriété également intéressante, elle est liée au décalage d'un signal, par exemple : soit $y(n)$ la version décalée en n_0 de $x(n)$

où : $y(n) = x(n - n_0)$. La transformée en z de $y(n)$ est [12] :

$$Y(z) = \sum_n y(n)z^{-n} = \sum_n y(n - n_0)z^{-n} = z^{-n_0} X(z) \quad (1.11)$$

Les filtres numériques utilisent souvent l'équation aux différences qui lie les suites d'entrée et celles de sortie. Elle se définit comme suit [12]:

$$y(n) = \sum_{i=0}^N a_i x(n-i) - \sum_{i=1}^N b_i y(n-i) \quad (1.12)$$

où N est l'ordre du filtre.

La fonction de transfert est obtenue en appliquant la transformée de z aux deux membres de l'équation aux différences. Elle a pour expression [12]:

$$H(z) = \frac{y(z)}{x(z)} = \frac{\sum_{i=0}^N a_i z^{-i}}{1 + \sum_{i=1}^N b_i z^{-i}} \quad (1.13)$$

Pour obtenir la réponse en fréquence, il suffit de remplacer dans H(z), la variable z par $e^{-j2\pi f}$. H(z) est une fonction rationnelle, il est possible de factoriser le numérateur N(z) et le dénominateur H(z), la fonction peut s'écrire sous la forme [12]:

$$H(z) = a_0 \frac{\prod_{i=1}^N (1 - Z_i z^{-1})}{\prod_{i=1}^N (1 - P_i z^{-1})} \quad (1.14)$$

a_0 est appelé un facteur d'échelle, Z_i représente les zéros de N(z) et P_i représente les pôles de D(z).

1.7.2 Structure de réalisation

Après avoir déterminé la fonction de transfert du filtre numérique. Une variété considérable de structures programmables de réalisation se présente. Le choix d'une structure doit être rigoureux. La structure choisie doit satisfaire les performances requises par le filtre.

Dans ce sous paragraphe, les structures les plus populaires seront présentées.

1.7.2.1 Structure directe

Les structures directes de type I et leurs transposées sont des structures canoniques. Ces structures seront développées au paragraphe 2.4.1 du chapitre 2

1.7.2.2 Structure en cascade

Pour obtenir une réalisation sous forme cascade, la fonction de transfert est décomposée en produit de fractions rationnelles $H_i(z)$ selon $H(z) = \prod_{i=1}^N H_i(z)$ comme l'indique la figure 7. Souvent, la structure cascade est obtenue par la simple mise en cascade du nombre voulu de cellules biquadratiques. Parfois, certaines de ces cellules peuvent être du premier ordre.

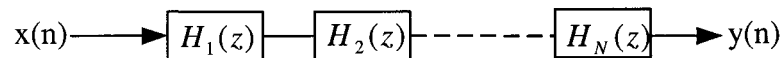


Figure 7 Structure cascade

1.7.2.3 Structure parallèle

Afin d'obtenir la forme parallèle d'un filtre numérique, la fonction de transfert est décomposée en somme partielle, $H_i(z)$ selon $H(z) = \sum_{i=1}^N H_i(z)$ comme l'indique la figure 8. Généralement, la structure parallèle n'est pas utilisée pour des filtres peu sélectifs.

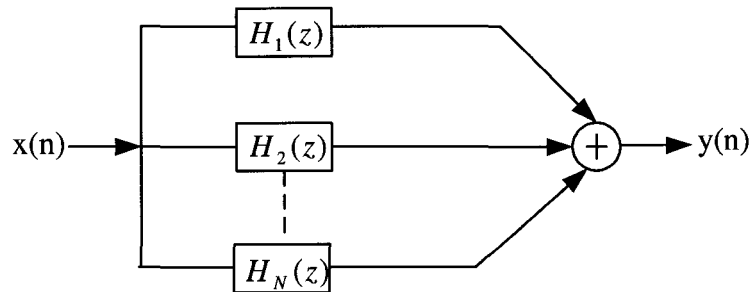


Figure 8 Structure parallèle

1.7.2.4 Structure polyphase

D'une façon générale, la structure polyphase comme indiquée à la figure 9 est utilisée pour réaliser les filtres RIF. Elle consiste à décomposer la fonction de transfert en une somme de M entités. La première entité ne contient que les indices de coefficients 0 modulo M , $h(0)$, $h(M)$, $h(2M)$... La seconde entité comprend uniquement les indices des coefficients 1 modulo M , $h(1)$, $h(M+1)$, $h(2M+1)$... En terme d'équation, $H(z)$ s'exprime sous la forme suivante [13]

$$H(z) = \sum_{l=0}^{M-1} z^{-l} E_l(Z^M) \quad (1.15)$$

où $E_l(z)$ représente la composante polyphase de la fonction de transfert $H(z)$ et a pour expression:

$$E_l(z) = \sum_{n=-\infty}^{+\infty} e(n)z^{-n} \quad (1.16)$$

avec

$$e_l(n) = h(Mn + l); \quad 0 \leq l \leq M - 1 \quad (1.17)$$

La structure polyphase est souvent utilisée dans le filtrage multcadence. Chaque structure présente des sensibilités et des performances de bruit différentes. Par conséquent, il est important de prendre en considération toutes les mesures nécessaires pour choisir adéquatement la réalisation du filtre. Cette réalisation doit satisfaire les spécifications requises par le filtre.

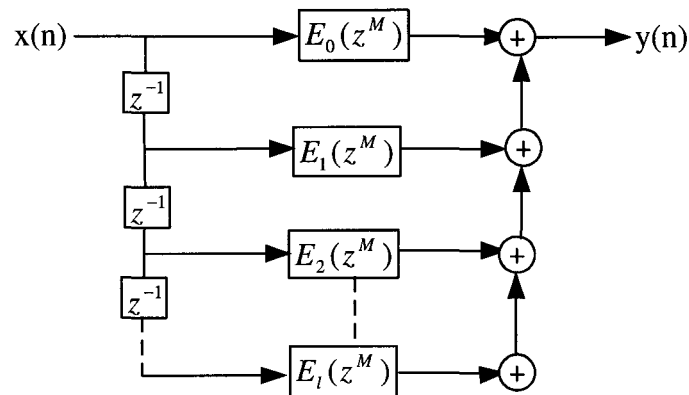


Figure 9 Structure polyphase

1.8 Filtres multcadence

La technique de filtrage multcadence devient de plus en plus populaire. Elle offre la possibilité d'avoir un signal avec différentes fréquences d'échantillonnage. Plusieurs applications requièrent un signal dont la fréquence d'échantillonnage est variable.

Notamment des systèmes en télécommunication qui transmettent ou reçoivent différents types de signaux. Par exemple, en audio, trois fréquences d'échantillonnage sont actuellement utilisées : 32 KHz pour la diffusion, 44.1 KHz pour le disque compact numérique, et 48 KHz dans la bande sonore numérique.

Afin d'obtenir des fréquences d'échantillonnage variables à différents étages, les systèmes multicadence utilisent des filtres décimateurs et des filtres interpolateurs.

1.8.1 Décimation

La décimation est définie comme la réduction de fréquence d'échantillonnage. Parfois, cette opération est appelée sous échantillonnage. La figure 10 représente un décimateur. Dans le cas où le rapport de réduction serait un entier R , la décimation consiste à ne garder qu'un échantillon sur R . En d'autres termes, la séquence de sortie est générée selon la relation suivante [13]:

$$y[n] = x[nR] \quad (1.18)$$

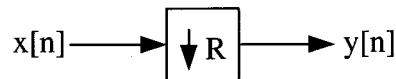


Figure 10 Représentation d'un décimateur

La condition du théorème de Nyquist doit être respectée. La figure 11 montre un exemple de décimation pour une séquence sinusoïdale $x[n]$ de longueur 100 avec un facteur de décimation $R = 2$. Il est possible que le rapport de décimation soit un nombre rationnel $\frac{I}{R}$. Dans ce cas, une augmentation de la fréquence est effectuée en premier

lieu pour avoir $f_e I$, suivie d'une décimation avec un rapport R . Ainsi le rapport $\frac{I}{R}$ est obtenu.

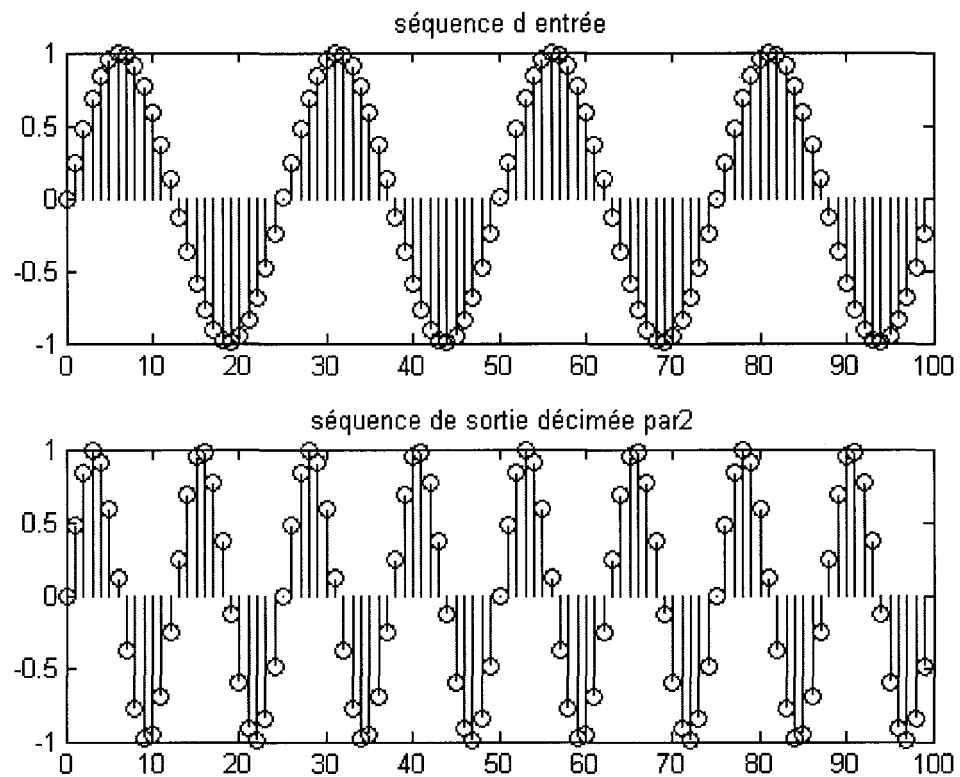


Figure 11 Exemple de décimation

L'opération de décimation est complétée en mettant un filtre passe-bas numérique de fonction de transfert $H(z)$ avant le décimateur comme l'indique la figure 12. On parle alors de filtre de décimation.

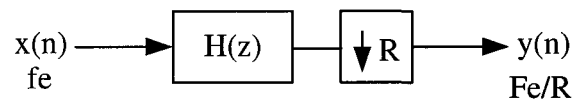


Figure 12 Filtre de décimation

Ce filtre assure une décimation du signal dans une bande limitée. Les bornes de cette bande sont déterminées par l'atténuation du filtre. Le module de la réponse fréquentielle d'un simple filtre de décimation est illustré à la figure 13

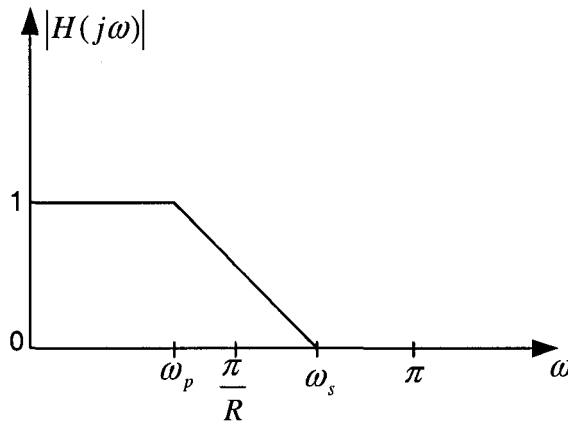


Figure 13 Module d'un simple filtre de décimation

1.8.2 Interpolation

L'interpolation consiste à augmenter la fréquence d'échantillonnage dans un rapport entier I . Parfois, elle est appelée suréchantillonnage. La figure 14 représente un interpolateur.



Figure 14 Représentation d'un interpolateur

Cette opération peut se faire en insérant $I-1$ valeurs nulles après chaque échantillon comme le montre l'exemple de la figure 15 pour une séquence sinusoïdale $x[n]$ de longueur 100 avec un facteur d'interpolation $I = 2$. En d'autres termes, la séquence de sortie est générée selon la relation [13]

$$y[n] = \begin{cases} x\left[\frac{n}{I}\right] & n = 0, \pm I, \pm 2I, \dots \\ 0 & \text{Autrement} \end{cases} \quad (1.19)$$

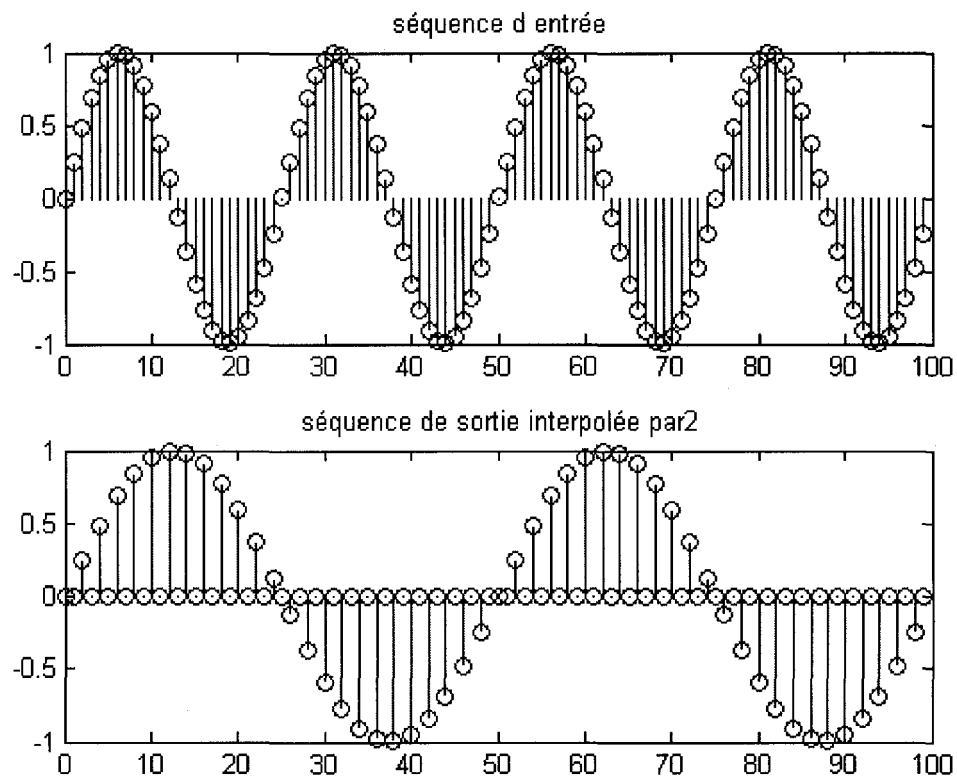


Figure 15 Exemple d'interpolation

Pour compléter le processus de l'interpolation, l'interpolateur doit être suivi d'un filtre passe-bas comme indiqué à la figure 16 pour supprimer les spectres dupliqués. On parle alors de filtres interpolateurs.

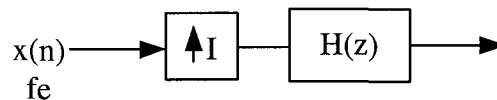


Figure 16 Filtre d'interpolation

Les filtres passe-bas de décimation ainsi que d'interpolation peuvent être conçus soit avec des filtres FIR ou RII. Le choix de l'un ou de l'autre dépend des exigences d'une application donnée. Dans le cas où l'application nécessiterait un ordre minimal du filtre, généralement l'utilisation d'un filtre RII est préconisée.

1.9 Conclusion

Dans ce premier chapitre, nous avons décrit les notions fondamentales couramment utilisées en traitement numérique du signal. Chaque outil de traitement de signal est conçu pour être adapté à une ou plusieurs classes de signaux.

La conception d'un filtre numérique est conditionnée par un choix approprié de la représentation du signal. Notamment, les méthodes de calcul des coefficients des filtres, ainsi que la structure de réalisation. Chaque élément doit être choisi adéquatement afin d'avoir les performances requises essentiellement pour le filtrage à haute vitesse.

CHAPITRE 2

FILTRES NUMÉRIQUES À RÉPONSE IMPULSIONNELLE INFINIE

2.1 Présentation générale des filtres RII

Les filtres numériques RII peuvent être utilisés pour la conception de filtres à hautes fréquences d'échantillonnage. Une étude détaillée de ce type de filtres est présentée dans ce chapitre.

2.1.1 Introduction

Les filtres numériques à Réponse Impulsionnelle Infinie (RII) sont des systèmes linéaires discrets invariants dans le temps. Chaque échantillon de sortie est égal à une combinaison linéaire des échantillons présents à l'entrée du filtre et des résultats précédemment déterminés en sortie du filtre.

Ils conservent une trace des signaux qui ont leur été appliqués pendant une durée infinie, ils sont donc à mémoire infinie. Une telle mémoire découle d'une boucle de rétroaction de la sortie sur l'entrée, d'où la dénomination courante de filtre récursif [9]. Le fait d'avoir cette réponse impulsionnelle infinie permet en général d'obtenir et à quantité de calcul équivalente des fonctions de filtrage beaucoup plus sélectives que des filtres à Réponse Impulsionnelle Finie (RIF). Cependant, la boucle de rétroaction complique l'étude des propriétés ainsi que la conception de ces filtres. La conception des filtres numériques RII consiste en trois étapes :

- 1- Élaboration des spécifications désirées du filtre, évidemment en fonction du problème technique.
- 2- Détermination des paramètres du filtre satisfaisant ces spécifications en utilisant les méthodes d'approximation appropriées.

3- Réalisation du filtre au moyen d'une structure adéquate.

2.1.2 Expressions générales

Afin de faciliter la notation, un certain nombre d'expressions générales caractérisant ces filtres est donné. Le filtre RII a pour équation aux différences l'expression suivante [12]:

$$y(n) = \sum_{l=0}^L a_l x(n-l) - \sum_{k=1}^K b_k y(n-k) \quad (2.1)$$

Ainsi, la fonction de transfert de ce filtre peut être exprimée sous la forme générale suivante [12] :

$$H(z) = \frac{\sum_{l=0}^L a_l z^{-l}}{1 + \sum_{k=1}^K b_k z^{-k}} \quad (2.2)$$

Les coefficients a_l, b_k sont habituellement des nombres réels, tandis que $H(z)$ est un complexe. Souvent les deux polynômes qui composent le numérateur et le dénominateur ont le même le degré. La réponse en fréquence du filtre RII s'exprime comme suit :

$$H(j\omega) = |H(j\omega)| e^{-j\varphi(j\omega)} \quad (2.3)$$

Les expressions du module, ainsi que le déphasage sont éventuellement déterminées à partir de l'expression de la fonction de transfert donnée par l'équation (2.2). Elles ont respectivement les formes suivantes :

$$|H(j\omega)|^2 = \left[H(z)H(z^{-1}) \right]_{z=e^{j\omega}} \quad (2.4)$$

$$\phi(j\omega) = -\frac{1}{2j} \ln \left[\frac{H(z)}{H(z^{-1})} \right]_{z=e^{j\omega}} \quad (2.5)$$

où $\omega = 2\pi f$.

La fonction de transfert peut aussi être exprimée autrement, celles ci servent essentiellement au choix de la réalisation de ces filtres, ainsi que le calcul des coefficients du filtre. Ainsi à partir de $H(z)$ en fonction de ses pôles et ses zéros, si les polynômes du quotient de $H(z)$ ont le même degré pair N ; $H(z)$ s'écrit alors sous la forme cascade suivante :

$$H(z) = a_0 \prod_{i=1}^{\frac{N}{2}} \frac{1 + a_1^i z^{-1} + a_2^i z^{-2}}{1 + b_1^i z^{-1} + b_2^i z^{-2}} \quad (2.6)$$

2.2 Méthode de calcul des coefficients des filtres RII

Habituellement, le calcul des coefficients des filtres RII n'est pas effectué d'une manière directe. Il se fait en ayant recours à une fonction modèle. Celle-ci est une fonction réelle définie sur l'axe des fréquences. Les fonctions modèles sont des fonctions connues pour leurs propriétés de sélectivité, telles que les fonctions de Butterworth, Bessel, Tchebycheff et les fonctions elliptiques.

Ces fonctions modèles peuvent être choisies comme étant le carré du module la fonction de transfert à obtenir. Cependant, un obstacle apparaît à leur utilisation au calcul des coefficients des filtres numériques, elles ne sont pas périodiques ; tandis que les fonctions à obtenir ont la période f_c . Alors il faut établir une correspondance entre l'axe

réel et $[0, f_e]$. Une telle correspondance est fournie par une transformation conforme dans le plan complexe qui doit respecter les contraintes suivantes :

- Transformer une fraction rationnelle de la variable complexe s en une fraction rationnelle de la variable complexe z .
- Conserver la stabilité.

Une transformation simple est obtenue en faisant correspondre un élément du plan s à l'élément du plan z en respectant la relation suivante [2]:

$$s = \frac{2}{T} \left(\frac{1 - Z^{-1}}{1 + Z^{-1}} \right) \quad (2.7)$$

où T est la période d'échantillonnage utilisée.

Cette transformation appelée transformation bilinéaire est l'une des premières méthodes envisagées pour le calcul des coefficients des filtres RII.

Deux types de fonctions modèles sont brièvement présentées pour calculer les coefficients des filtres RII. Les fonctions de Butterworth à cause de leur simplicité et les fonctions elliptiques qui sont plus efficaces.

2.2.1 Filtres de Butterworth

La fonction de Butterworth d'ordre n est définie par [12]:

$$|T(\omega)|^2 = \frac{1}{1 + \left(\frac{\omega}{\omega_c} \right)^{2n}} \quad (2.8)$$

où ω_c représente la variable pour laquelle $|T(j\omega)|$ prend la valeur $\frac{1}{\sqrt{2}}$.

En effectuant les transformations nécessaires, le filtre numérique d'ordre n impair correspondant est donné par [9]

$$H(z) = \frac{1+z^{-1}}{(1+u)+(1-u)z^{-1}} \prod_{k=1}^{\frac{n-1}{2}} a_0^k \frac{(1+z^{-1})^2}{1+b_1^k z^{-1}+b_2^k z^{-2}} \quad (2.9)$$

où
$$a_0^k = \frac{1}{1+u\alpha_k+u^2}, b_1^k = 2a_0^k(1-u^2), b_2^k = a_0^k(1-u\alpha_k+u^2) \quad (2.10)$$

où
$$u = \frac{1}{\operatorname{tg}(\pi f_c T)} \text{ et } \alpha_k = 2 \cos\left(\frac{\pi k}{n}\right) \quad (2.11)$$

Alors que pour n pair, $H(z)$ a pour expression :

$$H(z) = \prod_{k=1}^{\frac{n}{2}} a_0^k \frac{(1+z^{-1})^2}{1+b_1^k z^{-1}+b_2^k z^{-2}} \quad (2.12)$$

avec
$$\alpha_k = 2 \cos\left(\pi \frac{2k-1}{2n}\right) \quad (2.13)$$

L'ordre N du filtre est déterminé à partir de son gabarit. Soit à réaliser un filtre dont la réponse en fréquence est supérieure ou égale à $1-\delta_1$ dans la bande $[0, f_1]$ est inférieure ou égale à δ_2 dans la bande $\left[f_2, \frac{f_e}{2}\right]$. Ainsi N a pour expression [9] :

$$N \geq \frac{\log\left(\frac{1}{\delta_2 \sqrt{2\delta_1}}\right)}{\log(\operatorname{tg}(\pi f_2 T)) - \log(\operatorname{tg}(\pi f_1 T))} \quad (2.14)$$

où

$$\frac{1}{1+\left(\frac{\omega_1}{\omega_c}\right)^{2n}} \geq (1-\delta_1)^2 \text{ et } \frac{1}{1+\left(\frac{\omega_2}{\omega_c}\right)^{2n}} \leq \delta_2^2 \quad (2.15)$$

Quand la bande de transition est assez faible, l'ordre du filtre se simplifie et devient inversement proportionnel à la largeur de bande [9]. Il en résulte qu'en pratique la sélectivité du filtre est assez limitée.

2.2.2 Filtres elliptiques

Les filtres elliptiques présentent la bande de transition la plus faible ce qui fait ils sont optimaux pour un ordre N donné. La fonction modèle fait appel aux fonctions elliptiques. Elle est sous la forme suivante [2][9] :

$$T^2(u) = \frac{1}{1 + \varepsilon^2 \operatorname{sn}^2(u, k_1)} \quad (2.16)$$

où $y = \operatorname{sn}(u, k)$ est défini implicitement par la fonction elliptique incomplète de première espèce

$$u = \int_0^{\arcsin y} \frac{d\theta}{(1 - k^2 \sin^2 \theta)^{1/2}} \quad (2.17)$$

L'ordre N du filtre numérique correspondant se calcule aussi à partir de son gabarit est donné par [2][9]

$$N \cong 1.076 \log \left(\frac{2}{\delta_2 \sqrt{\delta_1}} \right) \log \left(\frac{f_e}{\Delta f} \frac{4}{\pi} \sin \left(2\pi \frac{f_1}{f_e} \right) \right) \quad (2.18)$$

où δ_1 représente l'ondulation dans la bande passante, δ_2 représente l'ondulation dans la bande coupée, f_1 est la fréquence de bande passante, f_2 est la fréquence de début de bande affaiblie, $\Delta f = f_1 - f_2$ représente la bande de transition et f_e est la fréquence d'échantillonnage.

La simplicité de calcul des filtres de Butterworth est leur force. Par contre, une recherche systématique du filtre optimal ne peut être assurée par ces filtres. En raison de leur sélectivité qui est beaucoup plus faible que celle que doit posséder un filtre optimal.. Le filtre elliptique nécessite un ordre deux fois plus faible qu'un filtre Butterworth, réduit la complexité des circuits dans la même proportion qu'un filtre Butterworth. Ce qui justifie le choix d'utiliser cette méthode afin de concevoir un filtre passe-bas RII avec des spécifications définies.

2.3 Conception d'un filtre RII passe- bas

L'exemple suivant illustre la conception d'un filtre passe-bas RII qui a une haute fréquence d'échantillonnage et une large bande passante .Ce filtre doit satisfaire aux spécifications suivantes :

Fréquence d'échantillonnage = 1 GHz

Fréquence de bande passante = 2 MHz

Fréquence de début de bande affaiblie = 4 MHz

Ondulation maximale dans la bande passante = 1 dB

Atténuation = 60 dB

2.3.1 Détermination des coefficients du filtre

Comme il a été déjà mentionné au paragraphe précédent, la méthode elliptique est optimale pour obtenir une bande de transition la plus faible. Elle permet d'obtenir un

filtre d'ordre minimal. En utilisant un programme de calcul conçu avec Matlab. Le filtre obtenu est d'ordre $N = 5$ et a pour coefficients les valeurs numériques suivantes:

Pour le numérateur :

$$a_0 = 0.04690946233447 \cdot 10^{-3}$$

$$a_1 = -0.14065378834332 \cdot 10^{-3} ; a_2 = 0.09374435095805 \cdot 10^{-3}$$

$$a_3 = 0.09374435095805 \cdot 10^{-3} ; a_4 = -0.14065378834332 \cdot 10^{-3} ; a_5 = 0.04690946233447 \cdot 10^{-3}$$

Pour le dénominateur :

$$b_1 = -4.98809940570866 ; b_2 = 9.95267574221165 ; b_3 = -9.92942871598824$$

$$b_4 = 4.95322784477243 ; b_5 = -0.98837546523727$$

La fonction de transfert est sous la forme générale :

$$H(Z) = \frac{N(Z)}{D(Z)} \quad (2.19)$$

où :

$$N(Z) = 46.90946233447 \cdot 10^{-6} * (1 - 2.99840973107818Z^{-1} + 1.99841026293676Z^{-2} + \\ -1.99841026293676Z^{-3} - 2.99840973107818Z^{-4} + Z^{-5})$$

$$D(Z) = 1 - 4.98809940570866Z^{-1} + 9.95267574221165 Z^{-2} - 9.92942871598824Z^{-3} + \\ 4.95322784477243Z^{-4} - 0.98837546523727Z^{-5}$$

2.3.2 Réponse fréquentielle

On sait que la réponse fréquentielle correspond à une transformation de Fourier d'une réponse impulsionnelle. Elle n'est définie que si le filtre est stable. La réponse en fréquence comprend le module et le déphasage. Il existe plusieurs structures de réalisation de la fonction de transfert des filtres RII.

Un aperçu de quelques structures les plus utilisées sera donné plus loin. Dans l'exemple de conception du filtre passe-bas du paragraphe 2.3.1, la structure canonique de type II a été utilisée.

2.3.2.1 Module

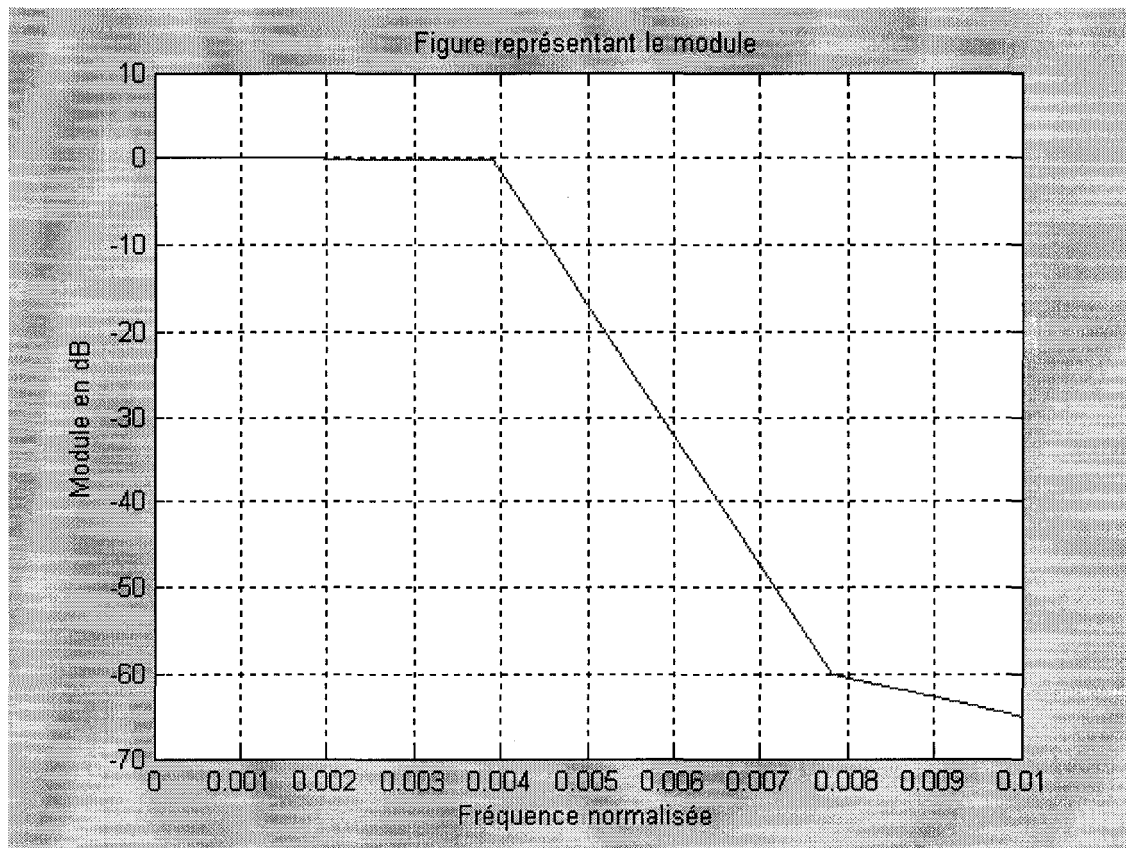


Figure 17 Module de la fonction de transfert $H(z)$ (2.19)

Le graphe du module de la réponse fréquentielle représenté sur la figure 17 correspond exactement aux spécifications du filtre passe-bas énoncé en 2.3.1 : Effectivement la largeur de bande est bien de 2 MHz avec une atténuation de 60 dB.

2.3.2.2 Déphasage

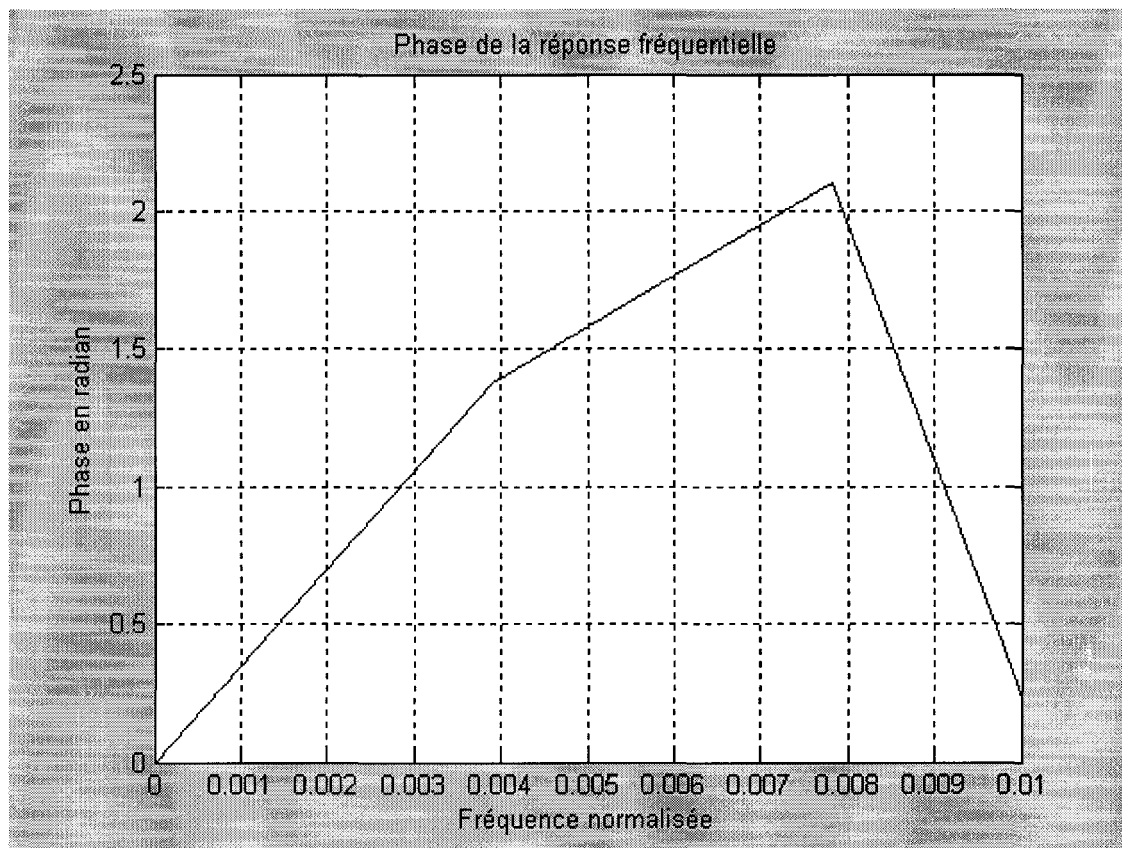


Figure 18 Phase de la fonction de transfert $H(z)$ (2.19)

Le déphasage de la réponse fréquentielle est quasi linéaire dans la bande passante. Cette propriété est intéressante pour plusieurs applications et rarement assurée par les filtres numériques RII. Au besoin, une correction pourrait être apportée à la partie non linéaire de la bande coupée [15] [16].

2.3.3 Concept de stabilité

La stabilité des filtres est importante du point de vue pratique. Un filtre numérique causale est stable si et seulement si tous les pôles et les zéros de ce filtre se trouvent à l'intérieur du cercle unité. Cette définition s'inscrit dans le cadre plus général que constitue la notion de stabilité. La vérification de la stabilité du filtre RII passe-bas est effectuée à l'aide d'un programme Matlab. Les valeurs respectives des pôles et des zéros du filtre passe-bas énoncé à la section 2.3.1 sont les suivantes :

$$P1 = 0.99902106836696 + 0.01248184462489i$$

$$P2 = 0.99902106836696 - 0.01248184462489i$$

$$P3 = 0.99706984273571 + 0.00832994900860i$$

$$P4 = 0.99706984273571 - 0.00832994900860i$$

$$P5 = 0.99591758350331$$

$$Z1 = -1.000000000000000$$

$$Z2 = 0.99944401847904 + 0.03334147453656i$$

$$Z3 = 0.99944401847904 - 0.03334147453656i$$

$$Z4 = 0.99976084706007 + 0.02186889773017i$$

$$Z5 = 0.99976084706007 - 0.02186889773017i$$

Voici les paramètres de test de la stabilité :

$$-0.99996037270789$$

$$0.99997877711752$$

$$-0.99997295915884$$

$$0.99994956672663$$

$$-0.98837546523727$$

Stable =

1

La figure 19 représente graphiquement la position des pôles et des zéros désignés respectivement par les symboles (\times) et (\circ). Tous les pôles ainsi que les zéros se trouvent à l'intérieur du cercle unité.

Cependant, même si la condition de stabilité est remplie, le filtre se trouve à la limite de la stabilité. Par conséquent, le filtre est fort sensible à toutes variations. Il devient alors nécessaire de prendre d'autres considérations, essentiellement vis à vis de la quantification des coefficients du filtre. Pour cette raison, et à titre d'exemple, si les coefficients sont quantifiés à 4 chiffres significatifs, le module de la réponse fréquentielle obtenu est représenté par la figure 20

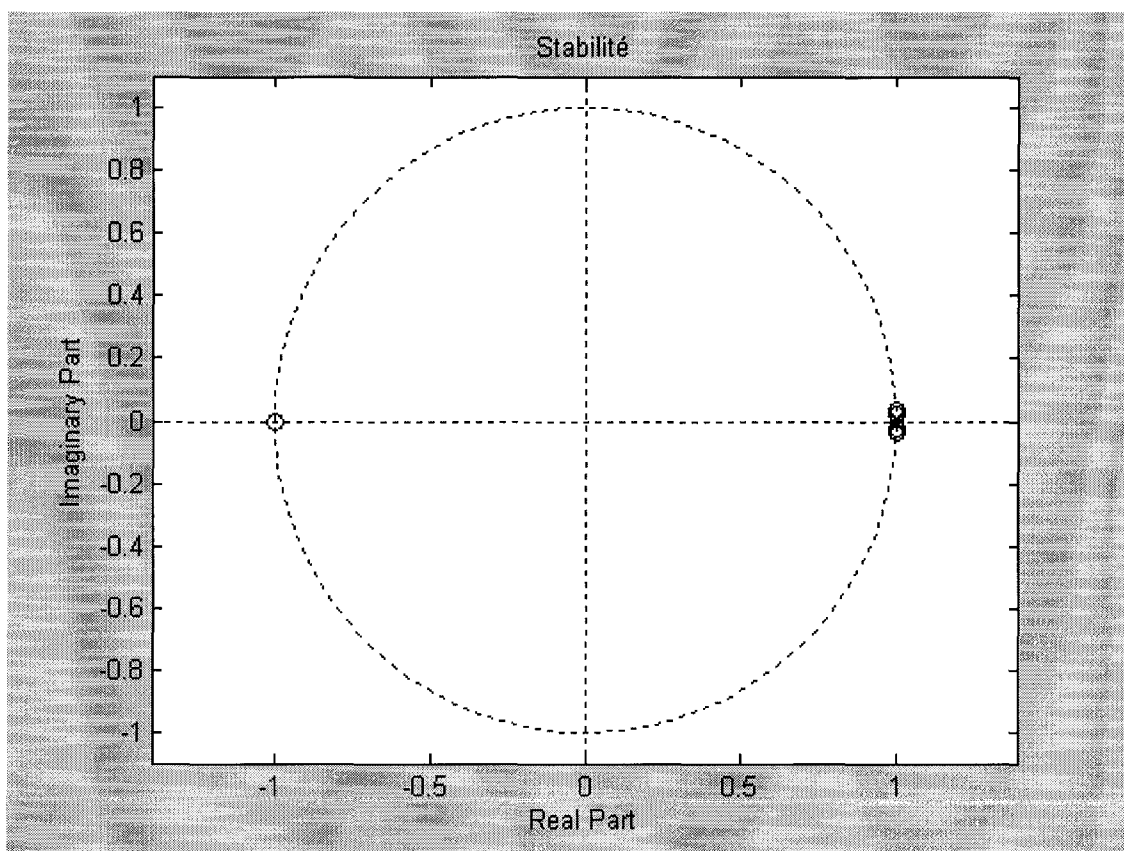


Figure 19 Stabilité de la fonction de transfert $H(z)$ (2.19)

Il est clair de par la figure 20 que les spécifications désirées du filtre sont loin d'être satisfaites. Ceci s'explique par le fait que les pôles de la fonction de transfert du filtre sont très proches du cercle unité. Ce qui rend le filtre très sensible et instable. Cela signifie que plus de quatre chiffres significatifs seront nécessaires afin d'atteindre les caractéristiques requises. Ce qui se traduit au niveau pratique par un certain nombre de bits. A titre indicatif, quatre chiffres significatifs nécessitent quinze bits, dont un bit de signe, et comme il a été déjà démontré que c'était très insuffisant, il faut seize chiffres pour avoir exactement les mêmes caractéristiques. Donc cinquante cinq bits pour chaque coefficient et par conséquent, la complexité augmente énormément.

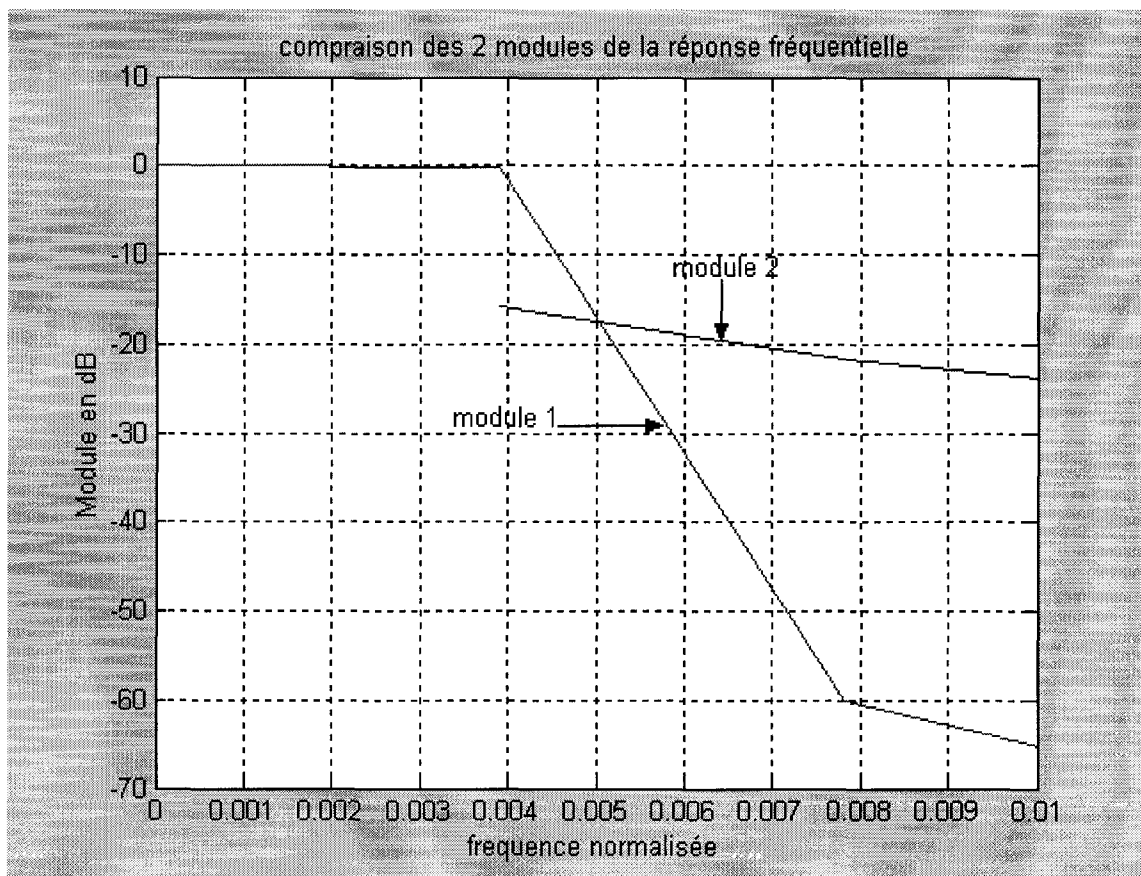


Figure 20 Comparaison du module (module1) avec tous les chiffres significatifs des coefficients et le module à quatre chiffres significatifs(module2)

2.4 Structures de réalisation des filtres RII

Les filtres numériques peuvent être réalisés par des circuits qui effectuent directement les opérations représentées dans l'expression de leur fonction de transfert. Le terme z^{-1} correspond à un retard d'une période d'échantillonnage réalisé par une mise en mémoire. Les coefficients à mettre en œuvre dans le circuit sont ceux de la fonction de transfert avec le même signe pour le numérateur et le signe contraire pour le dénominateur. Un aperçu sera donné uniquement pour les structures canoniques, étant donné que celles ci utilisent le minimum d'opérateurs élémentaires.

2.4.1 Structures directes

En général, les filtres RII peuvent être considérés comme une mise en cascade de filtres RIF et de filtres purement récurrents. Leur fonction de transfert est donnée par l'équation (1.13). Cette fonction de transfert peut être réalisée par une structure directe de type I où le dénominateur est calculé en premier, la structure est dite aussi D-N. Cette structure est illustrée à la figure 21.

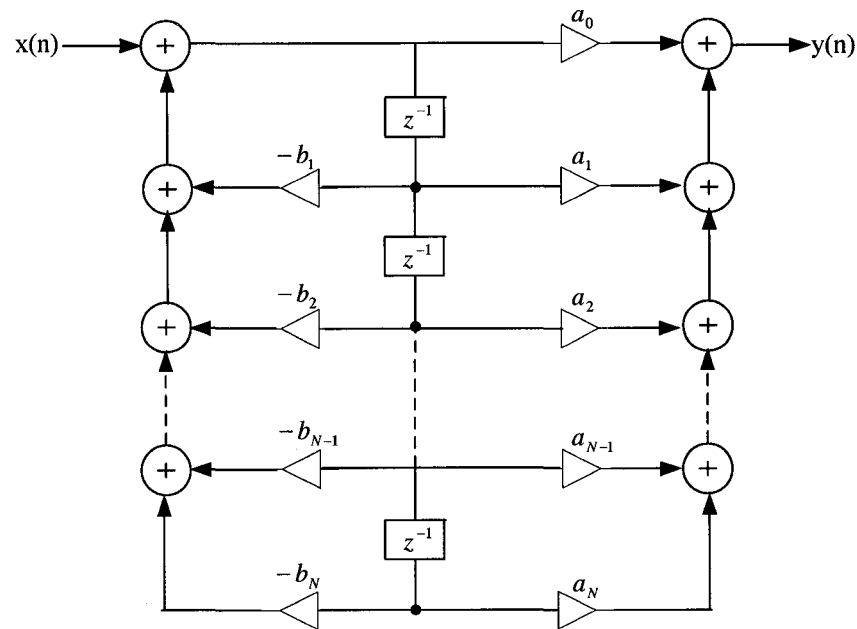


Figure 21 Structure directe de type I pour les filtres numériques RII

La structure directe de type II dite N-D représentée sur la figure 22 n'est autre que la transposée de la structure précédente. Cette structure présente une particularité intéressante : chaque nombre $y(n)$ ou $x(n)$ est multiplié par tous les coefficients, ce qui peut simplifier la mise en œuvre de la multiplication. C'est la raison pour laquelle elle a été utilisée dans l'exemple au paragraphe 2.3.1

Ces structures directes sont généralement peu utilisées en pratique à cause des difficultés de réalisation qui sont liées à la limitation du nombre de bits des coefficients, qui amènent à utiliser le plus souvent des structures composées

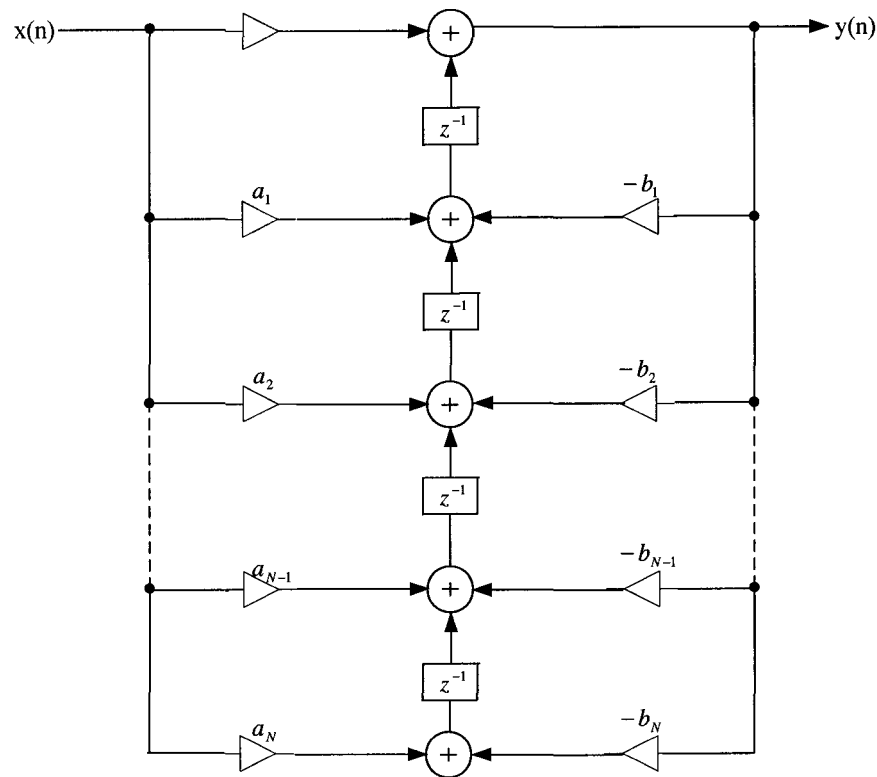


Figure 22 Structure directe de type II pour les filtres numériques RII

2.4.2 Limitation du nombre de bits des coefficients

La réalisation pratique des filtres numériques implique des opérations arithmétiques de multiplication. Ces opérations sont souvent le facteur le plus important pour la complexité, et causent fréquemment la limitation du nombre de bits. Il faut donc chercher le nombre minimal de bits satisfaisant les spécifications prédéfinies.

La limitation du nombre de bits du facteur d'échelle a_0 entraîne une modification du gain du filtre, mais la forme de la réponse fréquentielle reste la même.

Toutefois, il faut s'assurer que la représentation binaire de a_0 satisfait la contrainte du gain du filtre imposée par une application donnée. Par contre, la limitation de bits des autres coefficients du filtre modifie la fonction de transfert en introduisant des

polynômes parasites au numérateur noté $e_N(z)$ comme au dénominateur noté $e_D(z)$ la fonction de transfert devient [9]

$$H(z) = \frac{N(z) + e_N(z)}{D(z) + e_D(z)} \quad (2.20)$$

Ces polynômes parasites s'expriment en fonction des erreurs d'arrondi faites sur les coefficients. Leur expression est donnée par [9]

$$e_N(z) = \sum_{i=1}^N \delta a_i e^{-2\pi f_i} \quad (2.21)$$

$$e_D(z) = \sum_{i=1}^N \delta b_i e^{-2\pi f_i} \quad (2.22)$$

où $\delta a_i, \delta b_i$ représentent respectivement les erreurs d'arrondi du numérateur et du dénominateur du filtre.

Les conséquences de l'arrondi des coefficients peuvent être analysées séparément pour le numérateur et le dénominateur de la fonction de transfert, en considérant d'une part la bande affaiblie, et d'autre part la bande passante. En effet, les pôles déterminent la réponse du filtre en bande passante et les zéros en bande affaiblie. De ce fait, l'arrondi des coefficients du dénominateur peut être négligé en bande affaiblie, et l'arrondi des coefficients du numérateur peut être négligé en bande passante. De plus, quand la bande passante est étroite, les coefficients peuvent prendre des valeurs importantes et on peut ainsi écrire [9]

$$b_i \cong \frac{N!}{i!(N-i)!} \quad (2.23)$$

Dans ce cas, il faut un grand nombre de bits pour pouvoir représenter les valeurs des coefficients et obtenir en même temps une erreur de quantification très faible pour assurer les spécifications désirées. C'est pourquoi dans l'exemple du filtre du paragraphe 2.3.1 les résultats obtenus furent différents des résultats escomptés. En effet, ces résultats étaient loin des caractéristiques prédéfinies du filtre, lorsque les coefficients du filtre ont été approximés à quatre chiffres significatifs.

Une autre limitation intervient dans la réalisation des filtres numériques RII dans le cas des structures directes de type I ou II, est celle qui concerne la capacité interne des mémoires de données. Elle cause du bruit et peut aussi entraîner l'apparition d'auto oscillation de forte ou de faible amplitude et ce même en l'absence de signal à l'entrée du filtre.

2.5 Autres techniques

Une autre approche a été abordée. Elle consiste à utiliser des filtres numériques RII de second ordre en cascade avec le même nombre d'entrées et des sorties parallélisées. Deux techniques sont possibles. La première consiste à utiliser le filtre RII en structure pipeline. Cette structure est représentée à la figure 23 qui montre que les deux derniers résultats sont obtenus par une combinaison linéaire des derniers; dans ce cas, le nombre de multiplication des sorties non décimées demeure le même pour la forme non parallélisée, est égal à quatre, tandis que le filtre parallélisé nécessite $4P+2P$ multiplicateurs.

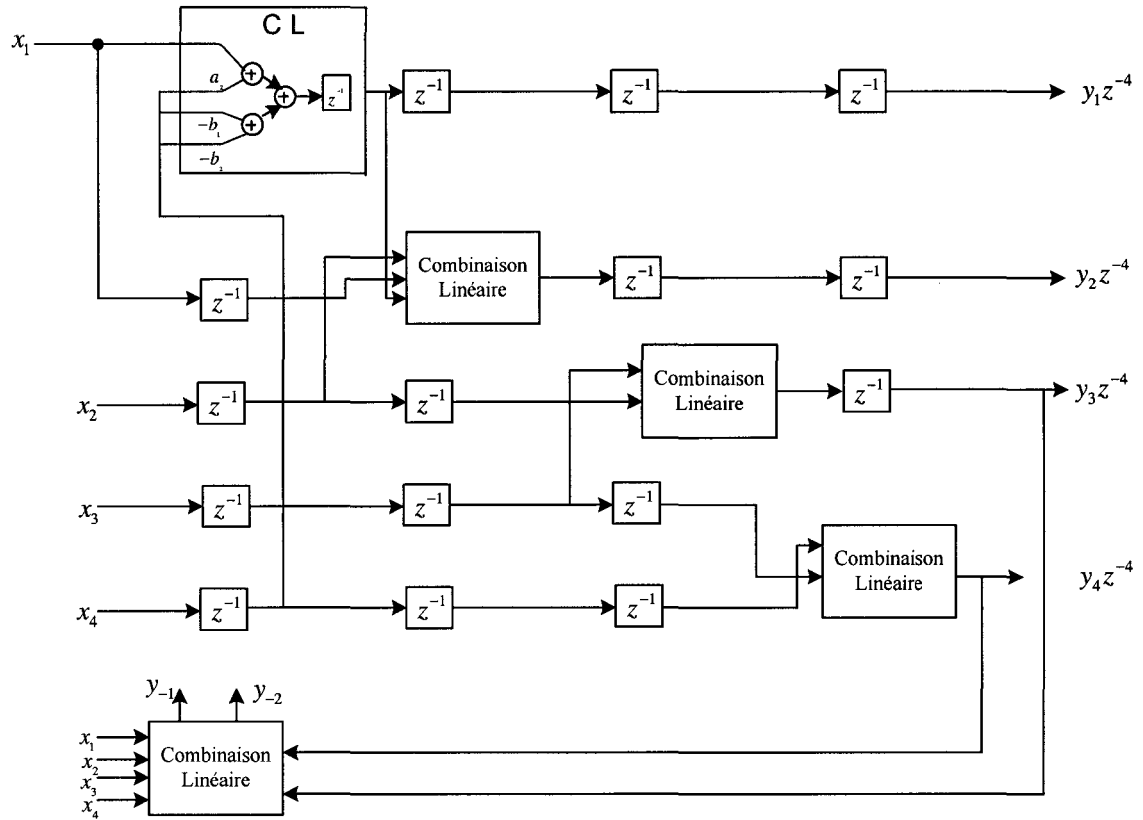


Figure 23 Schéma représentant le filtre RII en structure pipeline

La seconde technique consiste à exprimer P sorties comme étant une combinaison linéaire de P entrées, et les deux sorties précédentes restent mémorisées.

L'exemple suivant illustrera en détail la méthode.

$$\text{Soit : } H(z) = \frac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

L'équation aux différences est sous la forme suivante :

$$y(n) = b_0 x(n) + b_1 x(n-1) - a_1 y(n-1) - a_2 y(n-2)$$

Les trois premières sorties pour $P=3$ sont :

$$y(0) = b_0 x(0) + b_1 x(-1) - a_1 y(-1) - a_2 y(-2)$$

$$y(1) = b_0 x(1) + b_1 x(0) - a_1 (b_0 x(0) + b_1 x(-1) - a_1 y(-1) - a_2 y(-2)) - a_2 y(-1)$$

$$y(1) = b_0 x(1) + (b_1 - a_1 b_1) x(0) + a_1 b_1 x(-1) + (a_1^2 - a_2) y(-1) + a_1 a_2 y(-2)$$

$$y(2) = b_0 x(2) + (b_1 - a_1 b_1) x(1) + a_1 b_1 x(0) + (a_1^2 - a_2)(b_0 x(0) + b_1 x(-1) - a_1 y(-1) - a_2 y(-2)) + a_1 a_2 y(-1)$$

$x(1)$ reste mémorisé pour $y(1)$ et $y(2)$ pour la prochaine itération. La figure 2.8 l'illustre bien. Cette technique est simple, mais malheureusement elle est encombrante, surtout lorsque le circuit nécessite plus de parallélisme. Chaque sortie est une combinaison linéaire de $P+1$ entrée et les deux sorties précédentes doivent rester mémorisées. Ainsi chaque sortie est obtenue par $P+3$ multiplicateurs, et donc le filtre nécessite $(P+3)*P$ multiplicateur. Par conséquent, la complexité augmente inévitablement.

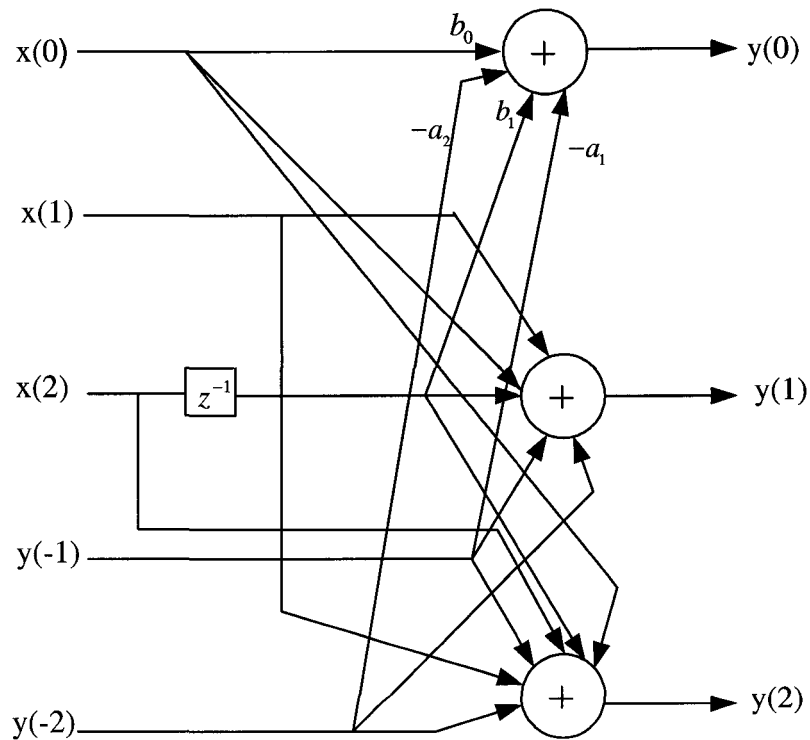


Figure 24 Schéma représentant le filtre RII en combinaison linéaire

2.6 Conclusion

Ce chapitre a été consacré à une étude des filtres numériques à impulsion infinie (RII). Cette étude présente en premier la conception du filtre passe-bas dont les spécifications sont définies. La méthode elliptique a été utilisée pour la détermination de l'ordre, des coefficients et la fonction de transfert du filtre. Le choix de cette méthode repose sur l'avantage qu'elle est optimale et permet d'obtenir l'ordre le plus petit que peut avoir un filtre pour des spécifications données, par rapport à d'autres méthodes.

La fonction de transfert a été réalisée avec une structure directe de type II. Une fois, la conception terminée, la stabilité de ce filtre a été étudiée. Effectivement, le test effectué a confirmé d'une part que le filtre est stable et d'autre part que les pôles sont très proches du cercle unité. Alors, une étude de la quantification des coefficients est devenue nécessaire pour vérifier si le filtre répond toujours aux spécifications requises.

Une autre approche a été aussi présentée. Elle consiste à utiliser des filtres numériques à impulsion infinie de second ordre en parallèle, soit en utilisant des filtres RII en pipeline soit en combinaison linéaire. L'emploi de l'une ou de l'autre de ces techniques présente des limitations qui proviennent principalement de l'utilisation des multiplicateurs. Ces multiplicateurs augmentent la complexité et ralentissent la vitesse du traitement. La complexité est un critère important pour le traitement numérique des signaux à haute vitesse.

Le résultat obtenu a été inattendu. Le filtre a perdu totalement ses caractéristiques. La réponse fréquentielle est devenue complètement différente de celle que devait avoir normalement le filtre. Ceci nous a amené à rechercher les causes de ce résultat surprenant. La limitation du nombre de bits des coefficients ainsi que la limitation de la capacité interne des mémoires en sont les principaux responsables.

Ainsi, l'ensemble des techniques décrites dans ce chapitre ne permet pas de répondre adéquatement à des contraintes très exigeantes requises pour le filtrage à haute fréquence.

Une autre technique économique et plus efficace permettant d'échantillonner à 1 GHz sera développée au cours du prochain chapitre.

CHAPITRE 3

FILTRES CIC DÉCIMATEURS

3.1 Introduction

Les résultats donnés par les filtres numériques RII indiquent clairement que la conception de filtres à hautes fréquences d'échantillonnage ne peut être effectuée par les méthodes décrites dans le chapitre précédent. Ceci à cause des problèmes rencontrés, découlant principalement de la quantification des coefficients des filtres. Par conséquent, une autre approche est utilisée qui ne souffre pas des limitations de la première approche.

Généralement, dans un filtre numérique le nombre de multiplication à faire par seconde est exprimé par $M_r = Kf_e$, où f_e représente la cadence à laquelle se font les calculs. Généralement f_e correspond à la fréquence d'échantillonnage et K est un paramètre qui dépend du type du filtre, de son ordre et de ses performances. Dans ce contexte, des filtres particulièrement efficaces ont été proposés, il s'agit des " Cascades Integrate and Comb filter", CIC [7]. L'absence de multiplicateurs et le non stockage des coefficients du filtre dans les filtres CIC, ainsi que la simplicité de leur implémentation présentent un grand avantage pour des applications exigeant des hautes fréquences d'échantillonnage; éventuellement les récepteurs radio, systèmes de communication à large bande,...etc.

3.2 Description générale

Les filtres CIC [7] sont des filtres multicaudence utilisés pour augmenter ou diminuer le rapport d'échantillonnage appelé respectivement interpolation ou décimation.

Une attention particulière sera accordée aux filtres CIC décimateurs.

La structure de base des filtres CIC décimateurs [7] indiquée à la figure 25 montre que ces filtres sont constitués essentiellement de deux blocs en cascade, le premier est composé d'un étage intégrateur numérique et le second est composé d'un étage différentiateur. Ces deux blocs sont séparés par un décimateur avec un facteur de décimation R . Cette structure de base peut avoir une structure équivalente illustrée à la figure 26, obtenue à l'aide des identités de Noble.

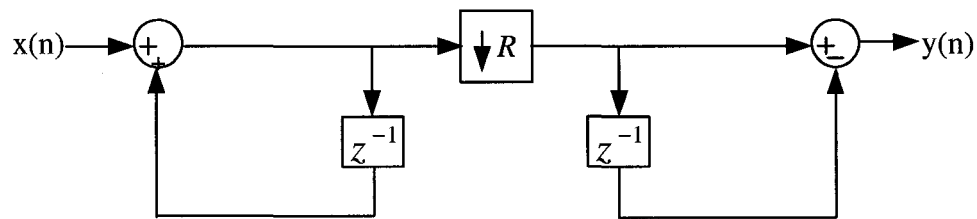


Figure 25 Structure de base d'un filtre CIC

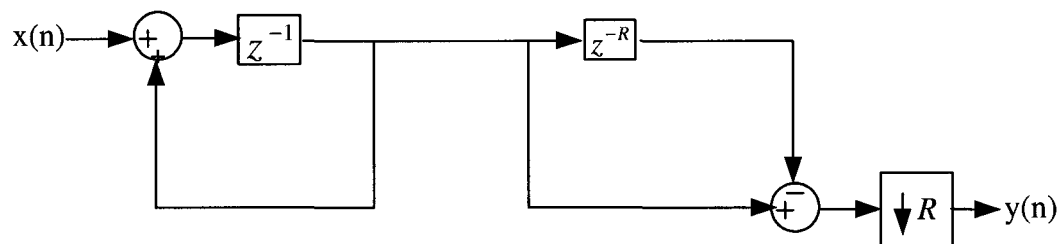


Figure 26 Circuit équivalent à la structure de base d'un filtre CIC

3.2.1 Bloc intégrateur

Chaque intégrateur est réalisé par un simple filtre à un seul pôle avec une contre réaction unitaire. Ces intégrateurs opèrent à la fréquence d'échantillonnage élevée f_s . Ils peuvent être considérés comme des accumulateurs.

La figure 27 représente un simple intégrateur, où $x(n)$ représente l'entrée, $y(n)$ la sortie et z^{-1} le délai.

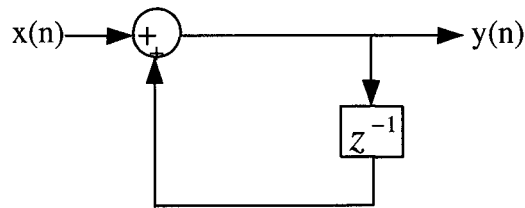


Figure 27 Schéma d'un simple intégrateur

L'équation aux différences de l'intégrateur numérique est donnée par :

$$y(n) = y(n-1) + x(n) \quad (3.1)$$

Et la fonction de transfert est sous la forme :

$$H_I(Z) = \frac{1}{1 - Z^{-1}} \quad (3.2)$$

En utilisant l'équation [17] appliquée à un système à un seul pôle, le module de sa réponse en fréquence est donné par la relation suivante:

$$\left| H_I(e^{j\omega}) \right|^2 = \frac{1}{2(1 - \cos \omega)} \quad (3.3)$$

Avec une approximation du premier ordre, la réponse en fréquence est celle d'un filtre passe-bas avec une atténuation de 20 dB par décade.

3.2.2 Bloc différentiateur

Les différentiateurs "Comb filter" fonctionnent généralement à fréquences d'échantillonnage plus basse f_e qui est égale à $\frac{f_s}{R}$, où R représente le facteur de décimation. Ces filtres sont aussi caractérisés par l'utilisation d'un paramètre de délai différentiel noté M qui sert à contrôler la position des zéros des fréquences de la réponse fréquentielle; en pratique souvent M prend la valeur 1 ou 2.

La figure 28 représente un différentiateur où les mêmes notations sont conservées.

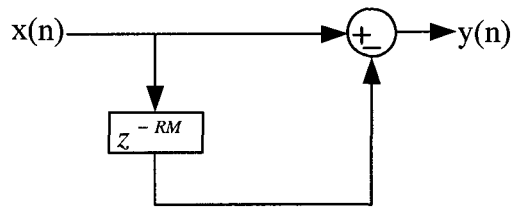


Figure 28 Schéma d'un simple différentiateur

L'équation aux différences est donnée par :

$$y(n) = x(n) - x(n - RM) \quad (3.4)$$

Et la fonction de transfert est sous la forme suivante :

$$H_c(Z) = 1 - Z^{-RM} \quad (3.5)$$

où R représente le facteur de décimation, et M représente le délai différentiel .

Le module de sa réponse en fréquence est donné par la relation suivante:

$$\left| H_c(e^{j\omega}) \right|^2 = 2(1 - \cos RM \omega) \quad (3.6)$$

La réponse en fréquence est celle d'un filtre passe-haut avec une atténuation de 20 dB par décade lorsque $RM\omega \prec 1 \text{ radian}$.

3.2.3 Filtre CIC complet

Le filtre CIC est ainsi constitué par le bloc de N étages intégrateurs opérant à f_e est suivi d'un décimateur qui réduit la fréquence de f_e à $\frac{f_s}{R}$, fréquence plus basse à laquelle le bloc de N étages différentiateurs fonctionne.

La fonction de transfert du filtre CIC décimateur a pour expression le produit de la fonction de transfert de l'intégrateur (3.2) par la fonction de transfert du différentiateur (3.4) soit :

$$H(Z) = H_I(Z) \times H_C(Z) = \left(\frac{1}{1-Z^{-1}} \right)^N \times (1-Z^{-RM})^N \quad (3.7)$$

$$H(Z) = \left(\frac{1-Z^{-RM}}{1-Z^{-1}} \right)^N$$

$H(z)$ peut être reformulée d'une autre façon en effectuant une simple division Euclidienne du numérateur par le dénominateur

$$\frac{1-Z^{-RM}}{1-Z^{-1}} = 1 + Z^{-1} + Z^{-2} + Z^{-3} + \dots + Z^{-RM+1} \quad (3.8)$$

L'expression 3.8 peut être appliquée à N étages et s'écrit sous une forme plus compacte :

$$H(Z) = \left[\sum_{K=0}^{RM-1} Z^{-K} \right]^N \quad (3.9)$$

Cette dernière forme exprime implicitement que le filtre CIC est équivalent à une série de N filtres FIR mis en cascade. L'intérêt de cette équivalence réside dans le fait que le filtre est stable et que la phase est linéaire. Ces propriétés sont primordiales pour le filtrage.

3.3 Théorie de conception des filtres CIC décimateurs

Cette section couvrira d'une façon approfondie et détaillée tous les aspects théoriques de la conception des filtres CIC décimateurs.

3.3.1 Paramètres de conception

Pour concevoir un filtre CIC décimateur, il faut d'abord déterminer l'ordre correspondant selon les spécifications techniques données. Ensuite, calculer la largeur des registres.

3.3.1.1 Ordre du filtre

L'ordre du filtre correspond au nombre d'étages N d'intégrateurs et de différentiateurs. Il est déterminé par les spécifications désirées du filtre, telles que la fréquence de coupure, l'atténuation. Pour ce faire, nous allons utiliser les tables 1 et de 2 données à la référence [7].

3.3.1.2 Largeur des registres

La sortie de chaque étage d'intégrateur accumule un nombre croissant de bits de l'étage précédent. Ceci peut causer un dépassement pour les registres internes, soit une perte d'information. Le nombre maximal de bits à mémoriser pour tout le filtre CIC est donné par [7] :

$$B_{\max} = N \log_2 RM + B_{in} - 1 \quad (3.10)$$

où N : nombre d'étages , R : taux de décimation , B_{in} : nombre de bits à l'entrée.

B_{\max} représente non seulement le nombre maximal de bits, mais aussi l'indice du bit le plus significatif (Most Significant Bit) pour la sortie totale du filtre. Dans la plupart des cas pratiques, B_{\max} est assez grand et peut entraîner un dépassement de la capacité des registres. Cependant, une procédure de troncation peut être utilisée à chaque étage et ainsi la largeur des registres est nettement réduite. Il est évident que cette troncation introduit une erreur à la sortie du filtre, même si elle s'effectue sur les bits les moins significatifs (Least significant Bit). La moyenne et la variance de l'erreur totale correspondent à la somme statistique de chaque source d'erreur, qui engendre un bruit blanc. De plus, la probabilité de distribution est uniforme et a pour expression [7] :

$$E_j = \begin{cases} 0, & \text{s'il n a pas de troncation} \\ 2^{B_j}, & \text{autrement} \end{cases} \quad (3.11)$$

où B_j représente le nombre de bit LSB à éliminer à chaque étage. Et du fait que l'erreur a une distribution de probabilité uniforme sa moyenne s'exprime donc comme suit [7] :

$$\mu_j = \begin{cases} \frac{1}{2} E_j, & \text{dans le cas d'une troncation} \\ 0, & \text{autrement} \end{cases} \quad (3.12)$$

Et sa variance a pour expression [7]:

$$\sigma_j^2 = \frac{1}{12} E_j^2 \quad (3.13)$$

La moyenne totale de l'erreur représente la somme statistique de chaque étage et s'exprime sous la forme suivante [7]:

$$\mu_{T_j} = \mu_j D_j \quad (3.14)$$

où D_j représente un gain de la moyenne et s'exprime comme suit:

$$D_j = \begin{cases} \sum_K h_j(K), & j = 1, 2, \dots, 2N \\ 1, & j = 2N + 1 \end{cases} \quad (3.15)$$

Cette expression peut être approximée sous la forme suivante [7]:

$$D_j = \begin{cases} (RM)^N, & j = 1 \\ 0, & j = 2, 3, \dots, 2N \\ 1, & j = 2N + 1 \end{cases} \quad (3.16)$$

La variance totale a pour expression [7]:

$$\sigma_{T_j}^2 = \sigma_j^2 F_j^2 \quad (3.17)$$

où F_j est un gain de la variance et a pour expression [7]:

$$F_j^2 = \begin{cases} \sum_K h_j^2(K), & j = 1, 2, \dots, 2N \\ 1, & j = 2N + 1 \end{cases} \quad (3.18)$$

La moyenne et la variance totales à la sortie du filtre dues à la troncation sont données respectivement par [7] :

$$\mu_T = \sum_{j=1}^{2N+1} \mu_{T_j} = \mu_{T_1} + \mu_{T_{2N+1}} \quad (3.19)$$

$$\sigma_T^2 = \sum_{j=1}^{2N+1} \sigma_{T_j}^2 \quad (3.20)$$

Cette troncation dépend essentiellement des paramètres N , R , M et le nombre total des bits (LSB) à éliminer est donné par la formule suivante [7]:

$$B_{2N+1} = B_{\max} - B_{out} + 1 \quad (3.21)$$

Pour l'étage j , compris entre 1 et $2N$, le nombre de bits LSB à éliminer à chaque étage est donné par [7]:

$$B_j = -\log_2 F_j + \log_2 \sigma_{T_{2N+1}} + \frac{1}{2} \log_2 \frac{6}{N} \quad (3.22)$$

où : F_j est donné par (3.18) , $\sigma_{T_{2N+1}}$ est donné par (3.17) et N représente l'ordre du filtre.

3.3.2 Réponse fréquentielle

Comme nous l'avons déjà mentionné dans la section précédente, le filtre CIC décimateur est constitué d'intégrateurs mis en cascade avec un décimateur et des différentiateurs. Reprenons l'expression de la fonction de transfert (3.5) :

$$H(z) = \left(\frac{1 - z^{-RM}}{1 - z^{-1}} \right)^N$$

La réponse en fréquence en amplitude $H(f)$ est obtenue à partir de l'expression (3.5) en remplaçant z par $e^{j(2\pi f/R)}$ ainsi l'expression du module est sous la forme suivante :

$$|H(f)| = \left| \frac{\sin(R.M.\pi.\frac{f}{f_e})}{\sin(\pi.\frac{f}{f_e})} \right|^N \quad (3.23)$$

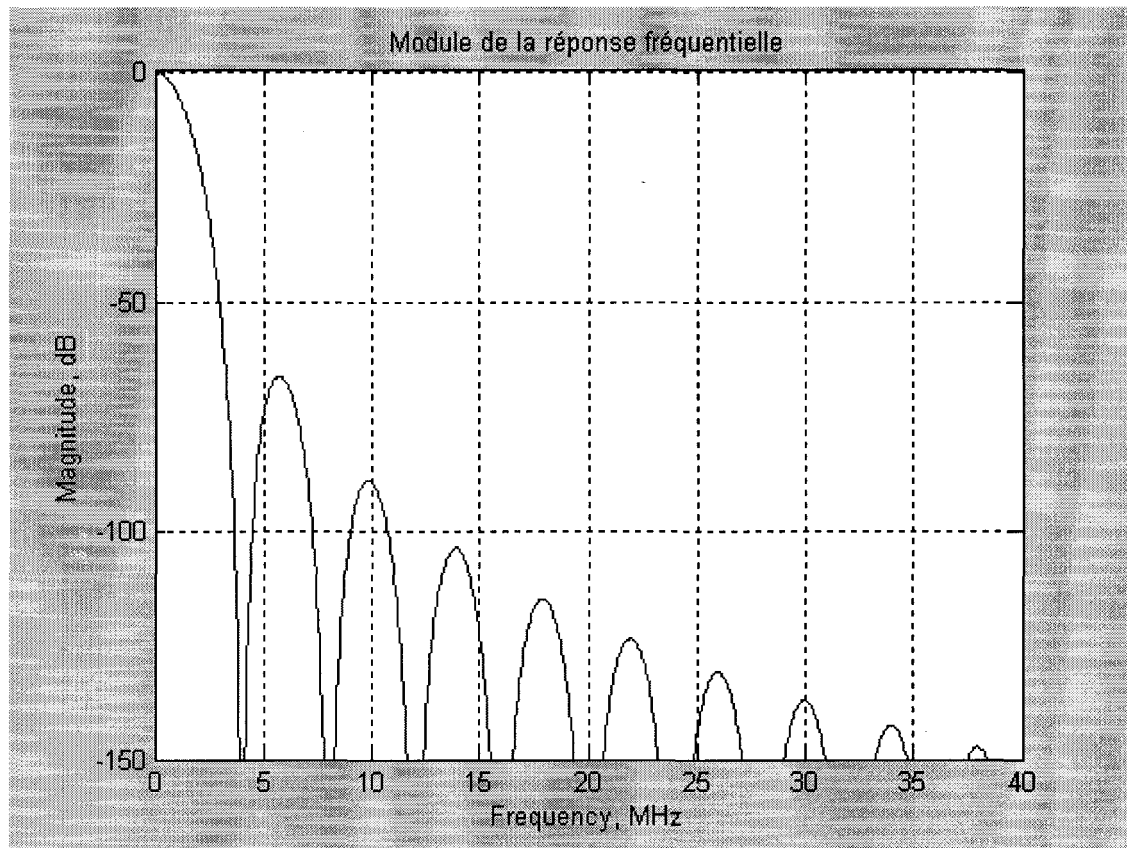


Figure 29 Module normalisé de la réponse en fréquence du filtre
CIC décimateur 3.2.2

La figure 29 montre un exemple du module de la réponse en fréquence d'un filtre décimateur d'ordre 5 ayant une fréquence d'échantillonnage égale à 1 GHz, un facteur de décimation R est égal à 250 et un délai différentiateur M est égal à 1.

3.4 Réalisation des filtres CIC décimateurs

D'une façon générale, les réalisations de filtres numériques ont été grandement liées au développement de la microélectronique, en particulier les circuits VLSI, ainsi que la technologie FPGA "Field Programmable Gate Array" [18].

Malgré le progrès et les performances qu'a connus la technologie FPGA au cours de ces dernières années, les opérations élémentaires sont encore limitées à environ 500 MHz. Alors que beaucoup d'applications, notamment en communication sans fil nécessitent de plus en plus un filtrage de signaux à haute fréquence qui peut atteindre facilement 1 GHz. Il faut donc trouver une architecture appropriée de réalisation de filtres numériques CIC décimateurs qui permettra aux circuits actuels d'implémenter un filtre de décimation à haute vitesse. Ainsi, différentes architectures sont proposées dans les sections suivantes.

3.4.1 Structure pipeline

C'est l'une des premières structures qui a été utilisée pour la réalisation des filtres CIC, en particulier les décimateurs [7]. Dans le mode pipeline, les divers blocs sont interconnectés en cascade comme le montre la figure 30. Des registres temporaires P0, P1, P2... sont nécessaires entre les divers étages du différentiateur.

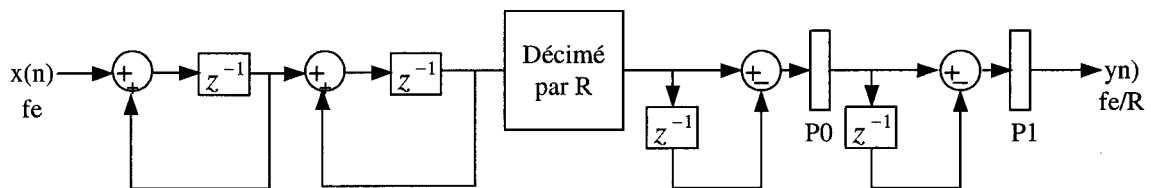


Figure 30 Structure pipeline d'un CIC décimateur à deux étages

L'avantage de cette technique réside dans sa facilité d'implémentation. Toutefois, elle reste limitée en terme de rapidité. En d'autres termes, son utilisation devient impossible pour des applications à haute vitesse.

3.4.2 Structure parallèle

La technique de parallélisme a pour objectif d'augmenter la rapidité d'exécution d'un processus. Elle reste aussi la structure permettant la plus grande vitesse de calcul pour les circuits arithmétiques, tels que les multiplieurs et les additionneurs qui constituent les opérations fondamentales pour les filtres numériques.

L'addition parallèle de deux nombres de n bits demande n cellules. Le temps nécessaire à l'exécution d'une addition parallèle dépend du temps de propagation de la retenue. Dans l'addition parallèle, le résultat reste disponible après la détection et la correction du dépassement. Tous les bits des opérations sont transmis simultanément aux opérateurs parallèles, ce qui réduit le temps de propagation. Donc, la vitesse de calcul augmente. En effet, la réalisation parallèle des filtres numériques devient de plus en plus importante. Elle est utilisée comme moyen d'accroître la vitesse de calcul dans les DSP [18]. Toutefois, il y a peu de méthodes qui ont été développées pour passer d'une forme directe à une forme parallèle [18].

A priori, la méthode pour paralléliser un filtre CIC décimateur est de considérer les entrées parallèles de l'intégrateur. La figure 31 montre bien le principe du parallélisme de: les 5 entrées X_1, X_2, \dots, X_5 sont générées simultanément et additionnées une à une de façon à obtenir une somme totale qui sera ensuite intégrée via l'intégrateur parallèle. De cette somme intégrée, il faut soustraire les valeurs X_1, X_2, \dots, X_5 afin de retrouver les images de X_1, X_2, \dots, X_5 . Des accumulateurs sont utilisés pour stocker les résultats intermédiaires. Ensuite vient le décimateur et le différentiateur.

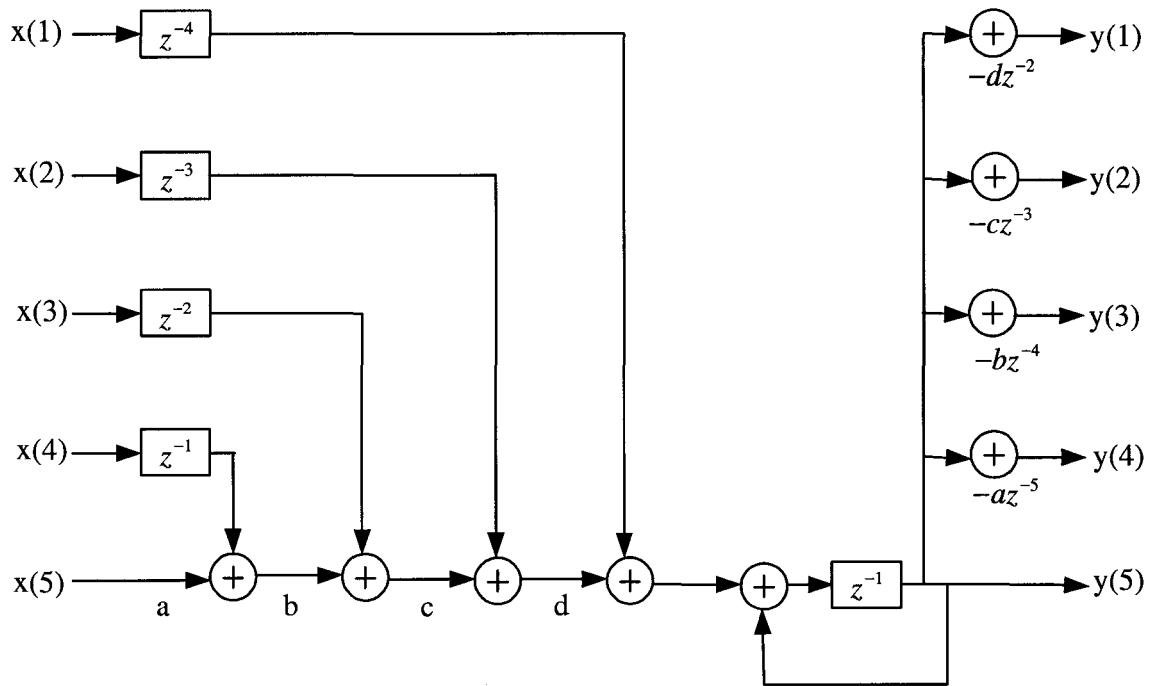


Figure 31 Structure parallèle de l'intégrateur du CIC

3.4.3 Structure parallélisée améliorée

C'est une structure qui utilise les additionneurs en arbre au lieu qu'ils soient en série comme le montre la figure 32 ainsi avec cette nouvelle structure, la latence est nettement réduite. Elle passe de $P+2$ à $\log_2(P)$. Le principe de cette structure est illustré à la figure 32.

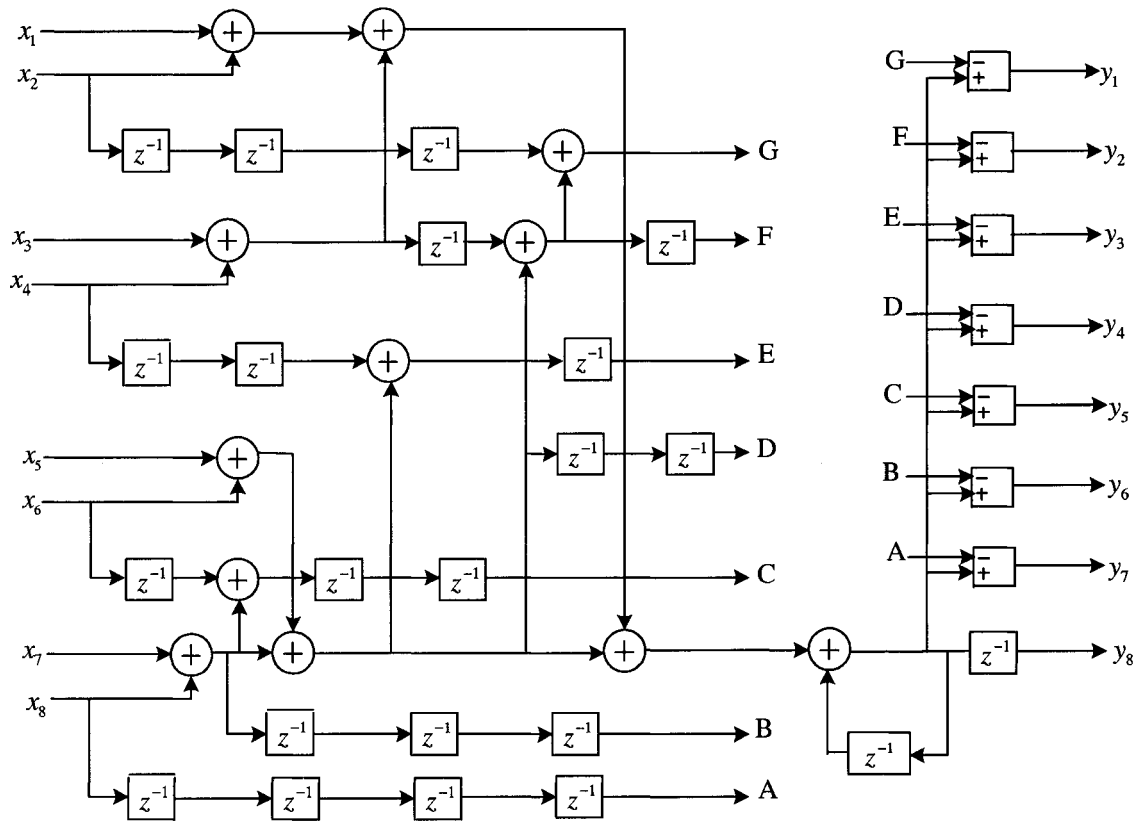


Figure 32 Structure parallélisée améliorée avec 8 entrées

L'utilisation de 8 entrées montre que l'implémentation de cette structure occuperait une surface CMOS assez grande et qui augmenterait avec l'augmentation du nombre de branches parallèles. Ce qui signifie une consommation de puissance importante. Malgré, la réduction de la complexité de cette structure, elle ne présente pas une solution économique.

3.4.4 Calcul de la complexité des deux structures

La complexité d'une architecture de traitement numérique de signal peut se définir comme la quantité de matériel nécessaire pour réaliser cette architecture, sous un ensemble de contraintes données. Elle peut varier dans des proportions considérables suivant l'architecture choisie. D'après la description des architectures faite dans le

paragraphe précédent. Les paramètres de complexité principaux qui peuvent être retenus sont les suivants :

- 1- Nombre d'additions ou additionneurs.
- 2- La taille des mémoires.

En considérant P le nombre de parallélisme à effectuer, l'addition se fait à un résultat précédent disponible dans un registre, la mise en mémoire de ce résultat est ensuite effectuée. De ce fait, pour l'architecture parallélisée représentée à la figure 30, la complexité est donnée par :

$$C=P(P+2)*b \quad (3.24)$$

où b représente le nombre de bits.

Pour la structure améliorée représentée sur la figure 32, la complexité s'écrit comme suit :

$$C = P \log_2(P) \times b \quad (3.25)$$

Il apparaît immédiatement que la complexité de la structure améliorée est nettement réduite ce qui engendre automatiquement l'amélioration de la latence. C'est un critère important pour le choix d'une architecture.

3.4.5 Structure parallèle utilisant la décomposition polyphase

La structure parallèle des filtres en particulier les filtres à réponse impulsionnelle finie (FIR) peut être réalisée en utilisant la décomposition polyphase [14].

Pour mieux illustrer cette approche, un exemple est donné : soit un filtre RIF d'ordre 6 dont la fonction de transfert est sous la forme suivante :

$$H(z) = h(0) + h(1)z^{-1} + h(2)z^{-2} + h(3)z^{-3} + h(4)z^{-4} + h(5)z^{-5} \quad (3.26)$$

Cette fonction de transfert peut être décomposée comme la somme de deux termes. Le premier ne contient que des coefficients pairs, le second est composé uniquement de coefficients impairs. En terme d'équation, l'équation 3.23 s'exprime comme suit :

$$\begin{aligned} H(z) &= (h(0) + h(2)z^{-2} + h(4)z^{-4}) \\ &\quad + (h(1)z^{-1} + h(3)z^{-3} + h(5)z^{-5}) \\ &= (h(0) + h(2)z^{-2} + h(4)z^{-4}) \\ &\quad + z^{-1}(h(1) + h(3)z^{-2} + h(5)z^{-4}) \end{aligned} \quad (3.27)$$

En utilisant les notations suivantes :

$$\begin{aligned} E_0(z) &= h(0) + h(2)z^{-1} + h(4)z^{-2} \\ E_1(z) &= h(1) + h(3)z^{-1} + h(5)z^{-2} \end{aligned} \quad (3.28)$$

La fonction $H(z)$ se récrit sous la forme:

$$h(z) = E_0(z^2) + z^{-1}E_1(z^2) \quad (3.29)$$

En analysant cette dernière équation (3.29), effectivement $H(z)$ peut être considérée comme deux fonctions de transfert de filtre FIR mises en parallèle. La figure 33 le montre bien. Cette structure est connue sous le nom de décomposition polyphase.

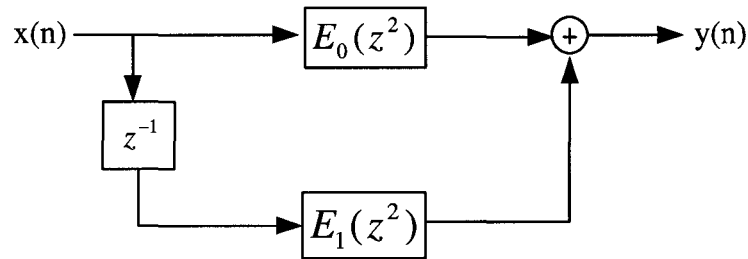


Figure 33 Structure polyphase du filtre FIR d'ordre 6 avec 2 branches

Dans le cas général, la décomposition polyphase avec M branches d'une fonction de transfert d'un filtre FIR d'ordre N est donnée par [13]

$$H(z) = \sum_{l=0}^{M-1} z^{-l} E_l(z^M) \quad (3.30)$$

où $E_l(z)$ représente la composante polyphase de la fonction de transfert $H(z)$ et a pour expression :

$$E_l(z) = \sum_{n=-\infty}^{+\infty} e_l(n) z^{-n} \quad (3.31)$$

où
$$e_l(n) = h(Mn + l); 0 \leq l \leq M - 1 \quad (3.32)$$

L'avantage de la décomposition polyphase des filtres RIF s'étend aux filtres RIF décimateurs. C'est une structure efficace, simple à appliquer pour la décimation.

On reprend le filtre RIF d'ordre 6 de l'exemple précédent avec une décimation égale à 2

La figure 34 montre la réalisation polyphase de ce filtre décimateur

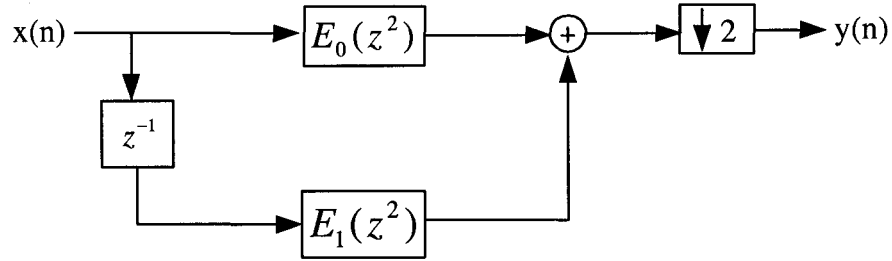


Figure 34 Structure polyphase du filtre FIR d'ordre 6

En appliquant l'identité de Noble [13] la figure 34 devient

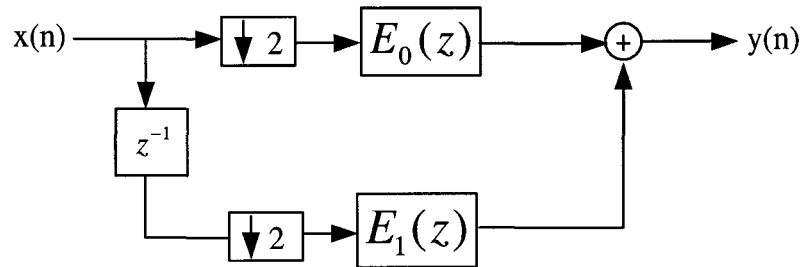


Figure 35 Structure polyphase du filtre FIR d'ordre 6

La structure polyphase réduit efficacement la complexité. Si on considère n_0, n_1 l'ordre respectivement pour $E_0(z)$ et $E_1(z)$ donc $N+1 = n_0 + n_1 + 2$. Ce qui signifie que $E_l(z)$ nécessite $n_l + 1$ multiplication et n_l addition, mais puisque $E_l(z)$ fonctionne à un taux de décimation bas, seulement $\frac{N+1}{2}$ multiplication par unité (MPUs) et $\frac{N}{2}$ addition par unité (APUs) sont requis [13]. En généralisant pour un filtre avec un facteur R de décimation, le filtre nécessite $\frac{N+1}{R}$ MPUs et $\frac{N}{R}$ APUs.

La décimation utilisant des filtres FIR présente de grands avantages. Toutes les opérations arithmétiques se font à la fréquence $\frac{F}{R}$ sans interpolation. L'application de la

décimation directe avec un taux d'échantillonnage élevé entraîne des distorsions dans la bande de base du signal [12].

Pour remédier à ce problème, une nouvelle technique de décimation a été utilisée. Cette nouvelle technique consiste à utiliser un filtre décimateur multiétage avec des taux de décimation différents [19]. En d'autres termes, le facteur de décimation total R sera le produit des facteurs R_1, R_2, \dots .

La figure 36 montre la réalisation de la technique de décimation multiétage utilisant des filtres CIC [19].

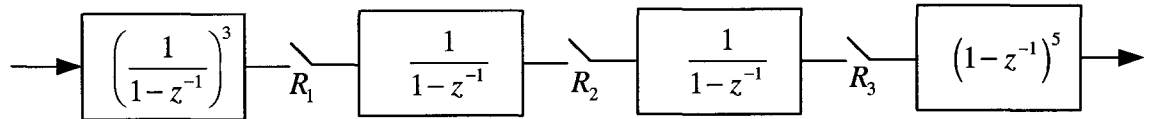


Figure 36 Décimation multiétage utilisant un filtre CIC [19]

Cette technique présente une efficacité et une simplicité d'implémentation qui seraient bénéfiques pour les filtres CIC décimateurs. Il faut donc trouver un moyen de décomposer le filtre CIC en polyphase.

La fonction de transfert d'un filtre CIC décimateur a été donnée par l'équation 3.6 suivante:

$$H(z) = \left(\frac{1 - z^{-R}}{1 - z^{-1}} \right)^N$$

Cette fonction de transfert d'ordre N avec un facteur de décimation R peut être réalisée d'une autre manière. On propose deux fonctions de transfert : la première fonction est d'ordre N_1 avec un facteur de décimation R_1 . La seconde fonction est d'ordre N_2 avec un facteur de décimation R_2 . Il est important de souligner que R doit être le produit de R_1 par R_2 .

$$H_p(z) = \left(\frac{1-z^{-R_1}}{1-z^{-1}} \right)^{N_1} \left(\frac{1-z^{-R_2}}{1-z^{-1}} \right)^{N_2} = H_1(z) * H_2(z^{R_1}) \quad (3.33)$$

En posant:

$$H_1(z) = \left(\frac{1-z^{-R_1}}{1-z^{-1}} \right)^{N_1} \quad (3.34)$$

$$H_2(z) = \left(\frac{1-z^{-R_2}}{1-z^{-1}} \right)^{N_2} \quad (3.35)$$

En utilisant l'équivalence 3.7 $H_1(z)$ peut s'exprimer sous la forme d'un RIF et a pour équation :

$$H_1(z) = \left(1 + z^{-1} + z^{-2} + \dots + z^{-(R_1-1)} \right)^{N_1} \quad (3.36)$$

Maintenant, $H_1(z)$ peut être décomposée en structure polyphase, pour obtenir l'expression :

$$H_1(z) = \sum_{M=0}^{R_1-1} z^{-M} E_M(z^{R_1}) \quad (3.37)$$

où $E_M(z)$ représente la composante polyphase, M est le nombre de branches.

Le choix de R_1 et N_1 doit être judicieux. Il faut que R_1 soit la plus petite valeur car elle est associée à la partie du filtre qui opère à la haute fréquence.

N_1 doit également être inférieur à N_2 et supérieur ou égal à 2 car le dernier étage doit avoir un ordre plus élevé pour avoir une meilleure atténuation [19].

Exemple :

On désire concevoir un filtre CIC avec les spécifications suivantes :

Fréquence d'échantillonnage= 1GHz

Bande passante inférieure ou égale à 1 MHz

Bande coupée= 4MHz

Atténuation=60 dB

Facteur de décimation= 250

En utilisant la table 1 et 2 [7] le filtre a un ordre 5

Le module de la réponse en fréquence est représenté à la figure 37. La fréquence de coupure est égale à 854 KHZ

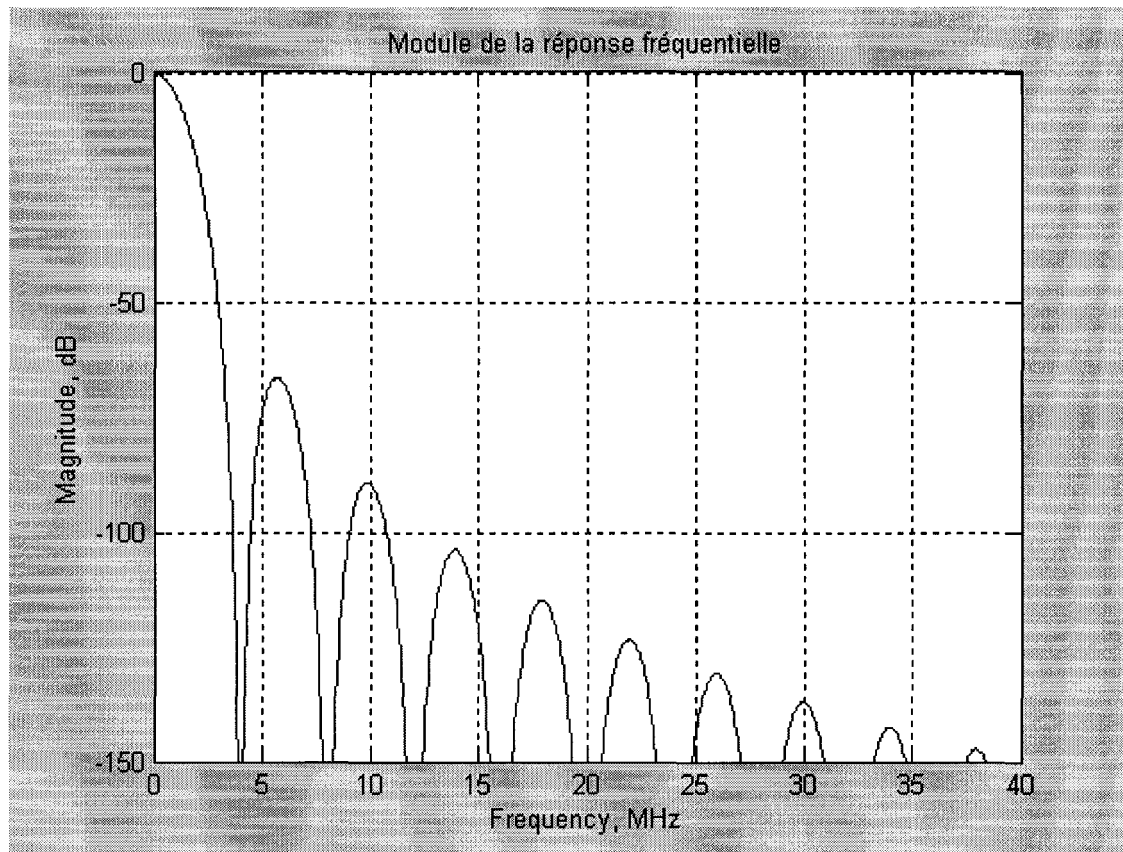


Figure 37 Module de la réponse fréquentielle du filtre CIC ($N=5$, $R=250$)

L'application de la nouvelle technique de décimation multiétage donne un facteur de décimation pour le premier étage $R_1=5$ avec un ordre $N_1=2$ et pour le second étage un facteur de décimation $R_2=50$ avec $N_2=3$.

Tout d'abord, on vérifie l'atténuation qui est contrôlée principalement par le second étage. Le module de la réponse en fréquence représenté à la figure 38 indique que l'atténuation est inférieure à l'atténuation fixée. Afin d'améliorer cette atténuation, un étage de filtre CIC peut être rajouté. Le module de la réponse en fréquence représenté à la figure 39 le montre bien. L'atténuation est effectivement améliorée. Elle a passé de 38 à 53 dB.

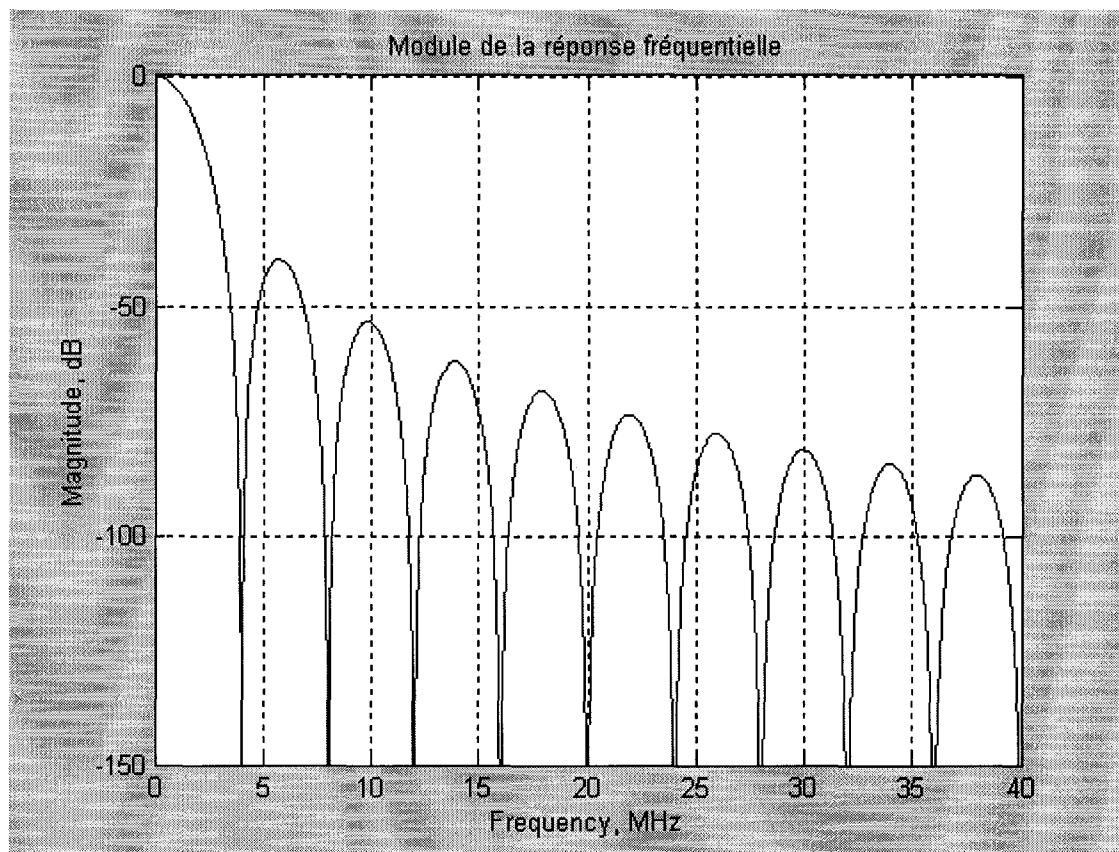


Figure 38 Module de la réponse fréquentielle du filtre CIC
 $(R_2=50, N_2=3)$

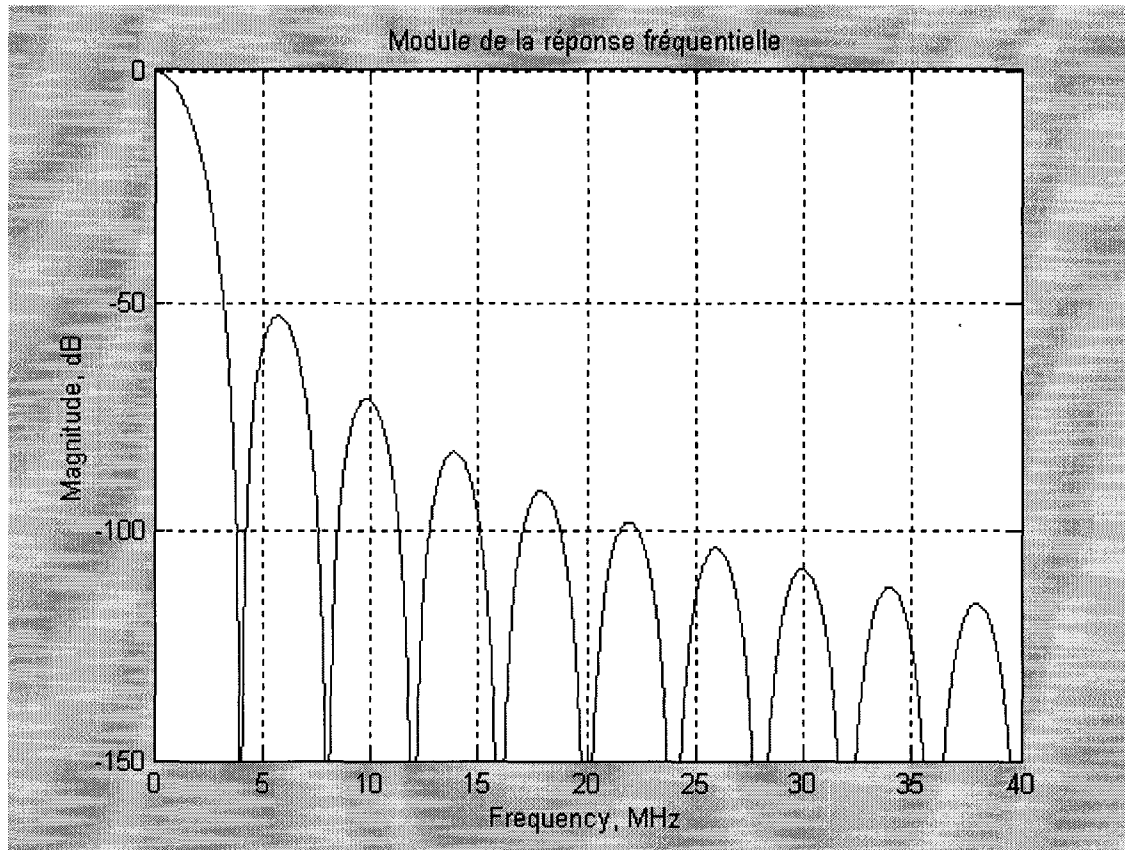
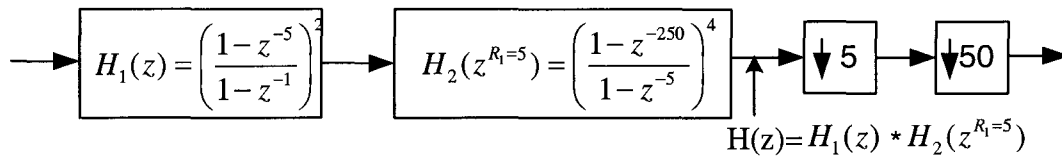


Figure 39 Module de la réponse fréquentielle du filtre CIC
($R_2=50$, $N_2=4$)

Ensuite, on analyse l'architecture globale du filtre



Le module de la réponse fréquentielle du premier étage illustré à la figure 40 montre un gain DC de 27.95 dB ce qui correspond exactement au calcul théorique

$$G_1 = 20 \log(5^2) = 27.96 \text{ dB}.$$

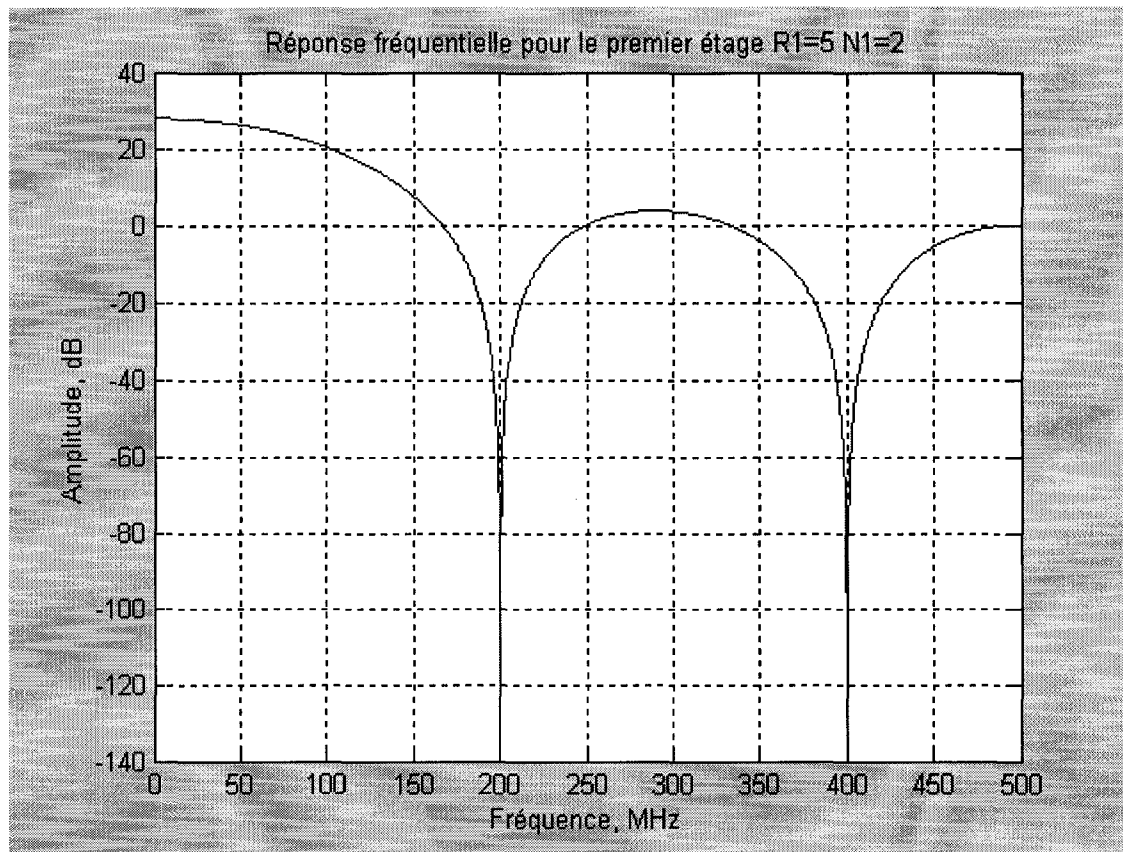


Figure 40 Module de la réponse fréquentielle du premier étage

Le second étage a pour fonction de transfert $H_2(z) = \left(\frac{1 - z^{-50}}{1 - z^{-1}} \right)^4$. Son module de la réponse en fréquence est présenté à la figure 41. Le calcul théorique du gain DC donne $G_2 = 20 \log(50^4) = 135.92$ dB ce qui est confirmé par le graphique 41.

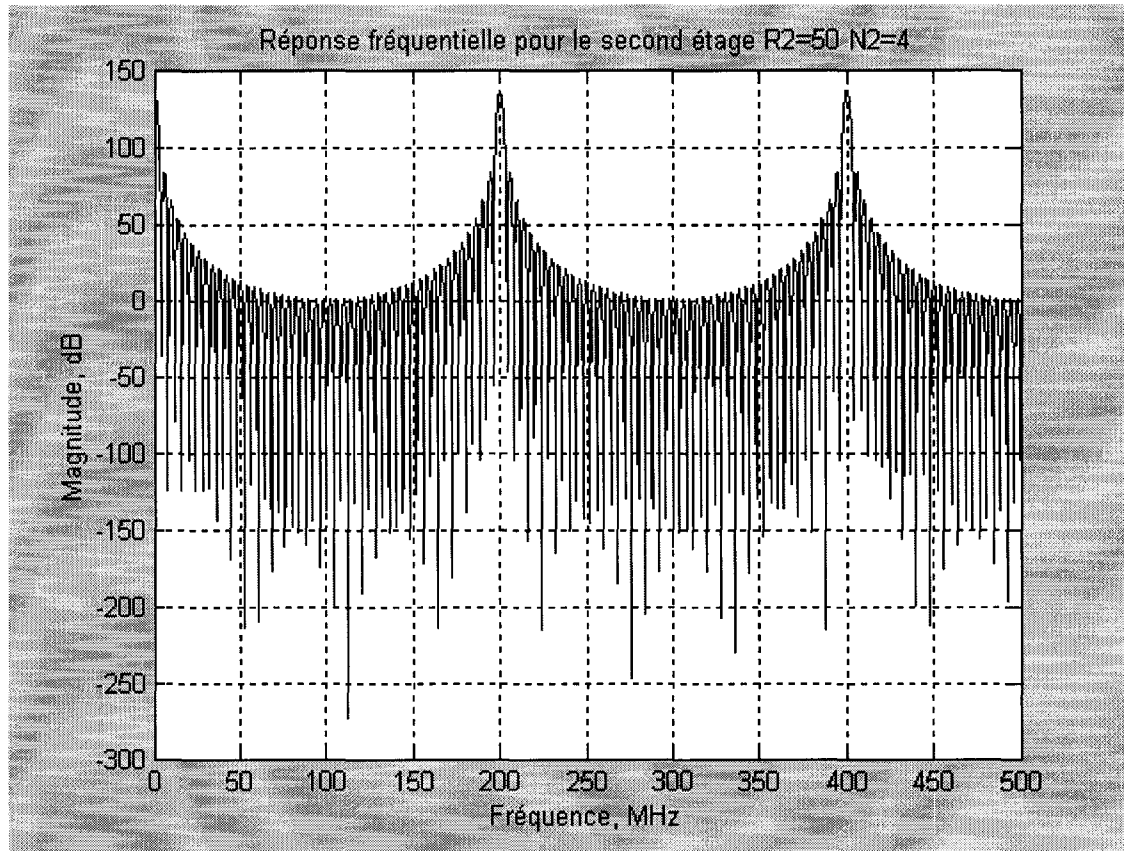


Figure 41 Module de la réponse fréquentielle du second étage

Le module de la réponse fréquentielle totale est présenté à la figure 42. Le gain DC calculé a pour valeur $G = G_1 + G_2 = 163.875$ dB ce qu'indique la figure 42. La reproduction d'images se fait effectivement à 200 MHz avec un gain de 75 dB.

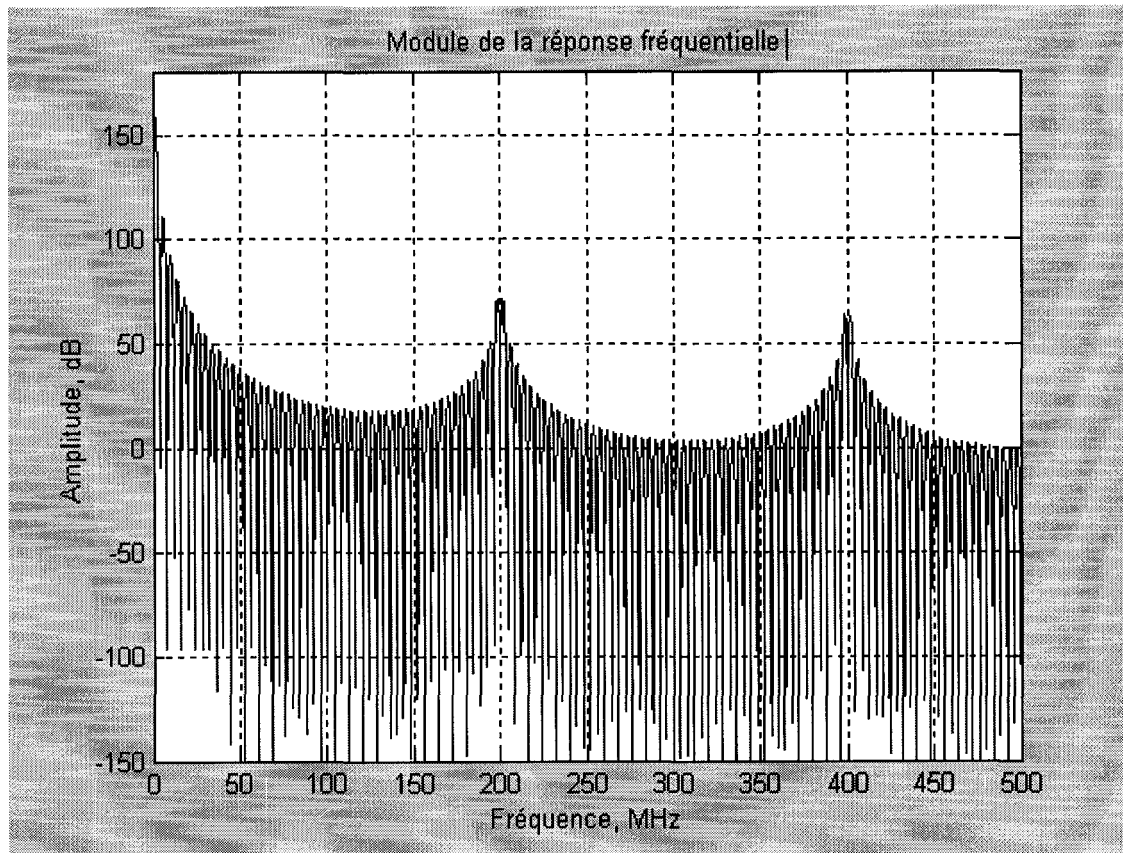


Figure 42 Module de la réponse fréquentielle du filtre au complet

Au niveau de la réalisation pratique, le premier étage utilise la décomposition polyphase comme façon de paralléliser les données à l'entrée et le second étage est un CIC standard. La nouvelle architecture proposée combine la décomposition polyphase et la décimation multiétage est efficace. Comme il a été mentionné, la décomposition polyphase réduit la complexité. De plus, avec l'utilisation juste des additionneurs la complexité est réduite d'avantage. Les composantes $E_M(z)$ sont réalisées avec des additionneurs et des délais.

3.5 Conclusion

Les détails de la théorie des filtres CIC ont fait l'objet de ce chapitre. D'abord, il fallait déterminer les paramètres de conception : à savoir l'ordre du filtre qui dépend des caractéristiques techniques désirées. Telles que la fréquence de coupure, l'atténuation... Puis vient la largeur des registres.

La structure de base des filtres CIC est en pipeline. L'utilisation de cette structure est impossible pour des applications à haute fréquence d'échantillonnage. Par conséquent, il fallait développer une architecture parallèle qui répond aux exigences d'un filtrage à haute vitesse. Ainsi, trois architectures ont été proposées. La première présente une structure parallèle. L'amélioration de cette structure constitue la deuxième architecture. En fin, une structure parallèle utilise la décomposition polyphase. Cette dernière structure sera retenue. Elle est avantageuse, car en plus de sa simplicité d'implémentation, elle permet d'atteindre des performances que les architectures classiques n'offrent pas.

CHAPITRE 4

CONCEPTION DÉTAILLÉE DES FILTRES CIC DÉCIMATEURS

Ce dernier chapitre présente la conception détaillée d'un filtre CIC à décimation parallélisée. L'architecture parallèle du filtre CIC utilisant la décomposition polyphase illustrée à la figure 43. L'étude théorique de cette architecture au cours du chapitre précédent a démontré son efficacité. Il reste à valider ses performances et à quantifier la complexité.

4.1 Application de l'architecture proposée

Afin d'évaluer les performances de l'architecture proposée. Un exemple de spécifications de filtre désirées illustre mieux cette nouvelle structure.

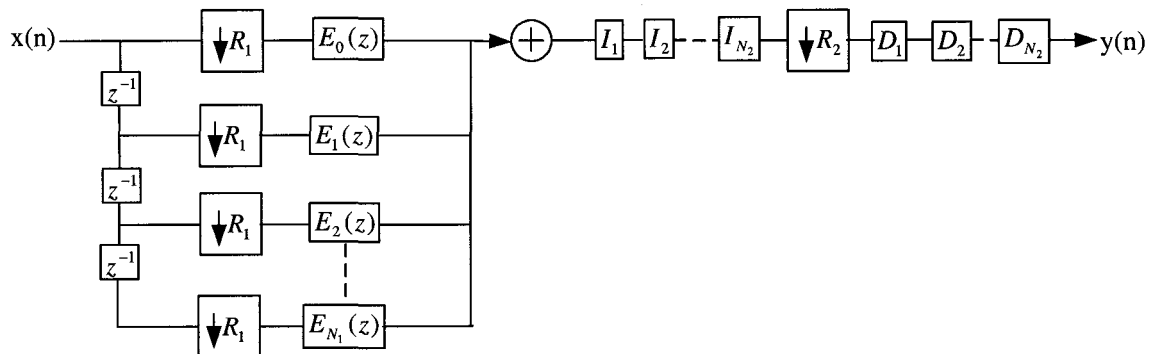


Figure 43 Structure parallèle d'un CIC utilisant la décomposition polyphase

On désire avoir les spécifications suivantes :

Nombre de bits à l'entrée : $B_{in} = 10$

Nombre de bits à la sortie : $B_{out} = 14$

Fréquence d'échantillonnage = 1GHz

Bande passante inférieure ou égale à 1 MHz

Bande coupée = 4 MHz

Atténuation = 60 dB

Facteur de décimation = 250

En premier lieu, il faut déterminer l'ordre N du filtre. En utilisant la table 1 [7], l'ordre N obtenu est égal à 5.

En appliquant le principe de décomposition de la nouvelle architecture proposée, le premier étage a $N_1 = 2$ et le facteur de décimation $R_1 = 5$. Il correspond à un filtre FIR dont la fonction de transfert est sous la forme suivante :

$$H_1(z) = \left(1 + z^{-1} + z^{-2} + z^{-3} + z^{-4}\right)^2 \quad (4.1)$$

La décomposition polyphase de $H_1(z)$ donne les composantes polyphases suivantes :

$$E_0(z) = 1 + 4z^{-1}$$

$$E_1(z) = 2 + 3z^{-1}$$

$$E_2(z) = 3 + 2z^{-1}$$

$$E_3(z) = 4 + z^{-1}$$

$$E_4(z) = 5$$

Par conséquent, l'expression de $H_1(z)$ devient :

$$H_1(z) = E_0(z^5) + z^{-1}E_1(z^5) + z^{-2}E_2(z^5) + z^{-3}E_3(z^5) + z^{-4}E_4(z^5) \quad (4.2)$$

La réalisation des composantes polyphases se fait uniquement par des additionneurs et des délais. La figure 44 montre un exemple de réalisation pour $E_0(z) = 1 + 4z^{-1}$

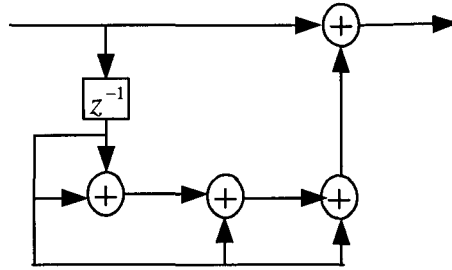


Figure 44 Exemple de réalisation pour $E_0(z)$

Les seconds étages correspondent aux filtres CIC standards. L'ordre du filtre CIC est $N_2 = 4$ au lieu de 3 pour avoir une meilleure atténuation comme il a été démontré au paragraphe 3.4.5. Le facteur de décimation $R_2 = 50$. Ainsi la fonction de transfert du filtre est sous la forme suivante :

$$H_2(z) = \left(\frac{1 - z^{-50}}{1 - z^{-1}} \right)^4 \quad (4.3)$$

En outre, il faut calculer le nombre de bits les moins significatifs (LSB) à rejeter. En procédant à la troncation comme il a été détaillé au sous paragraphe 3.3.1.2. Le tableau 4.1 résume les bits à éliminer.

La longueur maximale du registre est égale à 33 bits

Tableau I
Bits à rejeter pour chaque étage

	Étage N	B_{in}	Bits à rejeter	Total LSB à rejeter	Largeur de l'entrée	Largeur de sortie
Intégrateur	1	10	3	3	33	30
	2	30	5	8	30	25
	3	25	4	12	25	21
	4	21	2	14	21	19
Différentiateur	5	19	1	15	19	18
	6	18	1	16	18	17
	7	17	1	17	17	16
	8	16	2	19	16	14

4.2 Interprétation des résultats de simulation

Après avoir effectué tous les calculs nécessaires à la conception du filtre, la simulation de l'architecture proposée décrite à la figure 43 a été faite à l'aide du logiciel Simulink/Matlab. Le diagramme bloc du système simulé est illustré à la figure 45

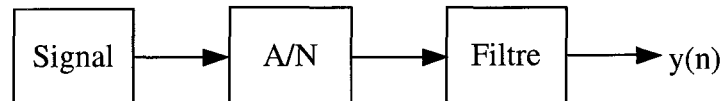


Figure 45 Diagramme bloc

Le signal d'entrée est à l'image d'un signal analogique réel. La conversion analogique à numérique se fait à l'aide d'un convertisseur analogique à numérique. Le signal de sortie de ce convertisseur n'est autre que le signal d'entrée quantifié par un nombre de bits. Le pas de quantification est déterminé par la représentation décimale.

Pour une représentation décimale entière, le pas de quantification est 1. Pour la représentation décimale fractionnelle, le pas de quantification est Δ et égale à :

$$\Delta = -2^{(nbits-1)} \quad (4.4)$$

La sortie est limitée par la représentation décimale. Pour la représentation décimale entière, la borne inférieure est :

$$B_{\text{inf}} = -2^{(nbits-1)} \quad (4.5)$$

La borne supérieure est :

$$B_{\text{sup}} = 2^{(nbits-1)} - 1 \quad (4.6)$$

Pour la représentation décimale fractionnelle les bornes inférieures et supérieures sont respectivement :

$$B_{\text{inf}} = -1 \quad (4.7)$$

$$B_{\text{sup}} = 2^{-(nbits-1)} \quad (4.8)$$

Le processus de quantification peut être considéré comme une troncation. Par conséquent, lors de l'implémentation de ce filtre, un choix adéquat du type de la représentation est important.

Ce signal quantifié constitue les données qui arrivent à l'entrée du filtre. Le premier étage de ce filtre fonctionne à un facteur de décimation très petit de valeur 5 et donc opère à une fréquence $\frac{f_c}{5}$. De plus, les données sont présentées d'une façon parallèle,

ce qui permet au filtre d'opérer sans difficulté à haute vitesse. Le choix de la valeur de N_1 et du facteur de décimation R_1 doit être approprié. La valeur de R_1 doit être le plus petit multiple commun de R et la valeur de N_1 doit être également petite. Ainsi le second étage qui est constitué par un simple CIC opère aisément. Les résultats de la simulation illustrés à la figure 46 le montrent bien.

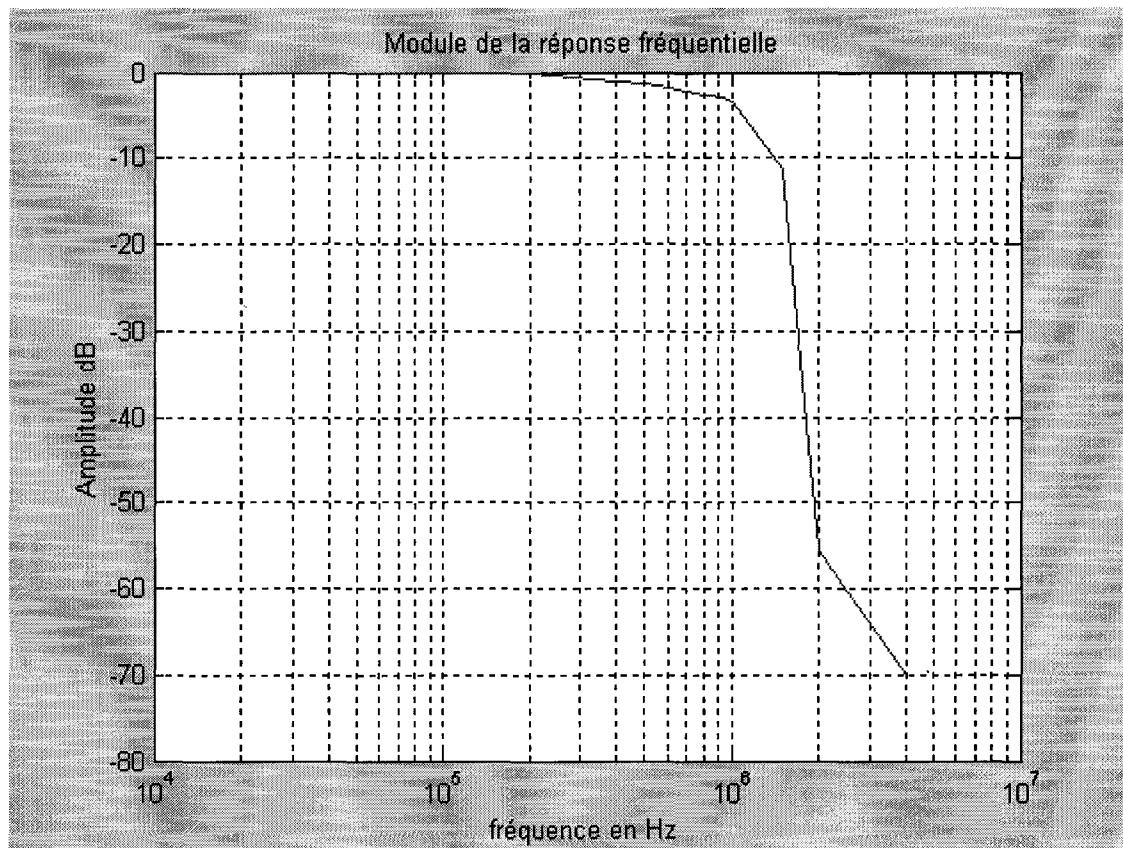


Figure 46 Module de la réponse fréquentielle

Ces résultats montrent que le filtre assure les spécifications désirées énoncées au paragraphe 4.1. En effet la fréquence de coupure est aux alentours de 925 KHz et la bande coupée est bien à 4 MHz avec une meilleure atténuation de 70 dB.

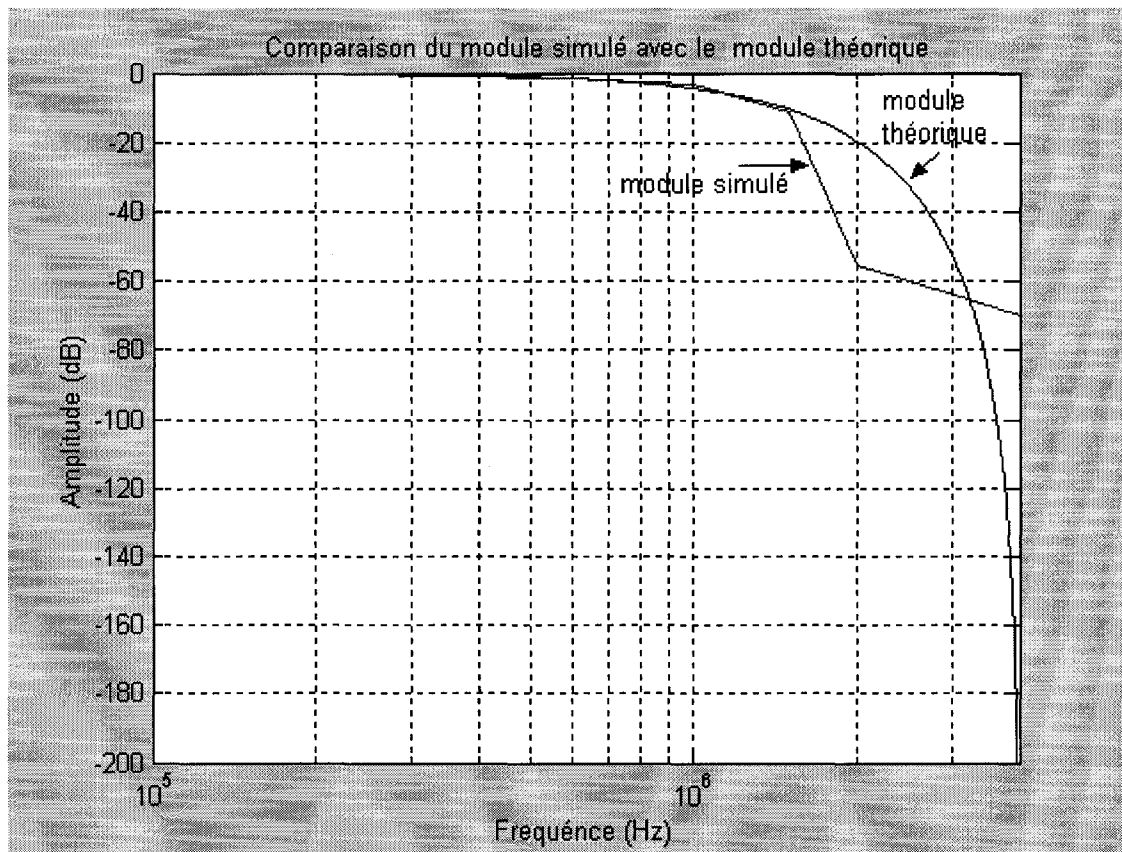


Figure 47 Comparaison du module simulé avec le module théorique

Le module de la réponse fréquentielle obtenu par simulation suit quasiment le module théorique du filtre CIC d'ordre 5 dans la bande passante. Toutefois, la fréquence de coupure est différente. Elle a une valeur théorique du filtre CIC d'ordre 5 de 854 KHz et une valeur obtenue par simulation de 925 KHz. Cette différence s'explique par l'utilisation de la décimation multiétage qui améliore la fréquence de coupure. À partir de 1.5 MHz qui représente le début de la bande de transition, le module obtenu par simulation devient différent du module théorique. La fréquence coupée a été définie à 4 MHz où l'atténuation a une valeur de 70 dB, ce qui représente une bonne atténuation.

4.3 Calcul de la complexité

Ce paragraphe présente une évaluation approximative de la complexité de réalisation en technologie FPGA, pour l'architecture proposée. Comme il a été déjà mentionné au paragraphe 3.4.4 la complexité dépend essentiellement du nombre d'additionneurs et de registres utilisés. Dans l'exemple simulé, la réalisation des composantes polyphases du filtre FIR qui constitue le premier étage du filtre se fait uniquement par des additionneurs et des délais comme l'indique l'exemple de la figure 4.3. Ainsi chaque composante utilise 4 additionneurs et un délai et il y a 5 branches parallèles. Le nombre de bits est égal à 10. Ainsi la complexité en termes de bits à mémoriser est égale à : $C = 10 \times 5 \times 5 = 250$. Ainsi, en comptant deux mémoires par slices on obtient 125 slices.

Le second étage n'est qu'un simple filtre CIC d'ordre 4. Il nécessite approximativement 215 slices. Donc la réalisation du filtre au complet demande approximativement $125 + 215$ soit 350 slices. Ce résultat montre que la complexité est faible. Toutefois, il est intéressant de valider la performance de la complexité de cette architecture par rapport aux autres architectures. Pour le calcul de la complexité du filtre CIC en utilisant l'architecture parallèle décrite au paragraphe 3.4.2 est donné par : la latence est égale à $P+2 = 12$. À chaque étage, le nombre de bits à l'entrée est réduit par élimination des bits les moins significatifs (LSB) selon [7]. Les bits à rejeter sont: 0,6,14,20,27,30,31,32,33,33. Donc les bits qui restent à l'entrée de chaque étage sont respectivement: 50,44,36,30,23,20,19,18,17,17. Le nombre de bits à l'entrée est égal à 10. Ainsi, la complexité totale a pour valeur:

$$C = 10 \times 12 \times (50 + 44 + 36 + 30 + 23) + (20 + 19 + 18 + 17 + 17) = 22051 \text{ soit } 11026 \text{ slices.}$$

Pour l'architecture parallèle améliorée, la latence $= \log_2(P) = \log_2(10) \cong 4$.

Le même nombre de bits est présent à l'entrée de chaque étage que dans l'architecture parallèle. Ainsi la valeur de la complexité totale est :

$$C = 10 \times 4 \times (50 + 44 + 36 + 30 + 23) + (20 + 19 + 18 + 17 + 17) = 7411 \text{ soit } 3705 \text{ slices.}$$

Le tableau II résume les résultats de cette comparaison

Tableau II

Comparaison de la complexité de différentes architectures

Architecture utilisée	Complexité en slices
Parallèle	11026
Parallèle améliorée	3085
utilisant la décomposition polyphase	350

Ce tableau confirme que l'architecture utilisant la décomposition polyphase réduit énormément la complexité par rapport à l'architecture parallèle et améliorée.

4.4 Conclusion

Dans ce chapitre, la simulation de l'architecture parallèle utilisant la décomposition polyphase a été présentée. Les résultats obtenus montrent que cette architecture assure de meilleures performances avec une complexité nettement réduite. Le filtre conçu opère sans difficulté à haute vitesse puisque la parallélisation est appliquée uniquement au premier étage. En plus, le facteur de décimation ainsi que l'ordre du filtre FIR sont très petits. L'utilisation du filtre FIR assure la stabilité ainsi que la linéarité de la phase. De plus, la décomposition polyphase permet de réduire la puissance consommée.

CONCLUSION GÉNÉRALE

La conception des filtres numériques utilisant des hautes fréquences d'échantillonnage n'est pas une tâche facile. Pour cela, le point de départ de cette recherche a été basé sur des techniques qui permettent d'accroître la vitesse de calcul et de diminuer le nombre d'opérations arithmétiques. D'où l'idée de chercher les structures parallèles ainsi que les structures qui utilisent moins ou pas d'opérations de multiplication, étant donné que la multiplication nécessite plus de capacité mémoire que l'addition. En tenant compte de ces critères, deux méthodes de conception de filtre multicadence ont été étudiées d'une façon détaillée.

La première méthode que nous avons considérée consiste à utiliser les filtres numériques RII. Ces filtres ont été conçus avec la méthode elliptique. L'utilisation de cette méthode optimale pour calculer les coefficients du filtre a permis d'avoir un petit ordre. Ainsi, les résultats obtenus montrent que ce filtre répond aux spécifications désirées. Sachant que les filtres RII présentent un risque d'instabilité, un test de vérification a été effectué. Le résultat de ce test a confirmé une stabilité critique. Ce qui a conduit à une étude de la quantification des coefficients du filtre. Cette étude a montré que le filtre était loin d'assurer les spécifications requises. La différence était énorme. Ce qui a éliminé la possibilité d'envisager des corrections.

Une autre possibilité a été abordée. Elle consiste à utiliser des filtres RII de second ordre en parallèle. Deux structures ont été envisagées. La première met en cascade les filtres RII. La seconde utilise la combinaison linéaire. Cette possibilité a été écartée à cause de ses limitations.

La deuxième méthode présentée dans ce travail consiste à utiliser les filtres CIC. Leur structure de base présente déjà un avantage qui réside dans le fait que leur réalisation ne nécessite aucune opération de multiplication. Toutefois, cette structure ne permet pas

d'atteindre de hautes fréquences. Cette limitation provient principalement des complications dans l'enchaînement des calculs qu'apporte la mise en cascade d'étages. Il fallait donc développer des structures parallèles surtout pour le premier étage. Ces considérations ont conduit à trois architectures parallèles. La première présente l'agencement des données en parallèle avec une latence de $P+2$. Cette architecture a été améliorée en effectuant une addition en arbre. Cette amélioration qui est la seconde architecture parallèle présentée a permis de réduire la latence à $\log_2(P)$. Malgré que la complexité ait été réduite cette structure reste encombrante et moins flexible. La troisième architecture parallèle considérée utilise la décomposition polyphase et la décimation multiétage. La décomposition polyphase permet de réduire la complexité. La décimation multiétage contrôle l'atténuation. Elle permet une bonne atténuation. La combinaison de ces performances fait la force de cette structure.

Les résultats de simulation obtenus confirment que cette architecture est efficace et économique. Elle présente de meilleures performances avec une complexité nettement réduite pour les hautes vitesses. Il faut souligner que l'implémentation de cette architecture est simple. Bien entendu, il faut que le choix des paramètres de conception N_1, R_1, N_2, R_2 soit adéquat comme il a été précisé par l'étude théorique. Le choix convenable de la représentation binaire joue aussi un rôle important pour avoir des meilleures performances.

En résumé, l'architecture que nous avons proposée a permis d'obtenir un filtre qui opère à haute fréquence d'échantillonnage de l'ordre de 1 GHz en utilisant les outils de programmation disponibles tels que FPGA. Ainsi, ces filtres peuvent être utilisés efficacement pour une démodulation quadratique ou pour des modulateurs sigma-delta, ainsi que pour plusieurs applications qui nécessitent une haute vitesse de traitement. Ainsi, nous pensons avoir atteint notre objectif initial.

BIBLIOGRAPHIE

- [1] Chandrakasan, A.P., Brodersan, R.W. (Apr. 1995). Minimizing power consumption in digital CMOS circuits Proceedings of the IEEE , 83(4), 498 –523.
- [2] Rabiner, L., Gold, B. (c1975). Theory and application of digital signal processing (Prentice-Hall).
- [3] Crochiere, R.,L Rabiner, (Oct 1975). Optimum FIR digital filter implementations for decimation, interpolation, and narrow-band filtering. Acoustics, Speech, and Signal Processing 23(5), 444 –456.
- [4] Rabiner, L., Crochiere, R. (Oct 1975). A novel implementation for narrow-band FIR digital filters. Acoustics, Speech, and Signal Processing, 23(5), 457 –464.
- [5] Peled, A., Bede Liu. (Dec 1974). A new hardware realization of digital filters. Acoustics, Speech, and Signal Processing , 22(6), 456 –462.
- [6] Goodman, D., Carey.M. (Apr 1977). Nine digital filters for decimation and interpolation. Acoustics, Speech, and Signal Processing , 25(2), 121-126.
- [7] Hogenauer, E. (Apr 1981). An economical class of digital filters for decimation and interpolation. Acoustics, Speech, and Signal Processing, 29(2), 155-162.
- [8] Alan V. Oppenheim, Ronald W. Schaffer with John R. Buck. (c1999). Discrete-time signal processing (Prentice-Hall 2nd ed).
- [9] Bellanger, M. (c1976). Traitement numérique du signal : théorie et pratique (Masson 5^{ème} éd).
- [10] Kunt, Murat., Granlund, Goesta. (c1991). Traitement de l'information (Presses polytechniques et universitaires romandes).
- [11] Boite, R., Leich, H. (c1990). Les filtres numériques analyse et synthèse des filtres unidimensionnels (Massons 3e éd. rév. et augm)
- [12] Proakis, J. G., Manolakis, Dimitris G. (1996). Digital signal processing : principes, algorithms, and applications (Prentice-Hall 3rd ed).
- [13] Vaidyanathan, P. P. (1993). Multirate systems and filter banks (PTR Prentice Hall).

- [14] Mitra, Sanjit Kumar. (1998). Digital signal processing: a computer-based approach. (McGraw-Hill).
- [15] Krukowski, A., Kale, I. (July 2001.). Polyphase Filter Design with Reduced Phase Non-Linearity, presented at 5th WSES/IEEE World Multiconference on Circuits, Systems, Communications & Computers CSCC 2001, Rethymnon, Crete (Greece)
- [16] Ekiz, O., Constantinides, A.G. (June 2000). Phase linearity in polyphase filters. IEEE International Conference, 1(5-9), 321 –323.
- [17] Brambilla, M., Guidi, D., Liberali, V. (29 Sept-2 Oct 1999). High speed FIR filters for digital decimation. Integrated Circuits and Systems Design, 1999. Proceedings. XII Symposium, 124 -127.
- [18] Krukowski, A., Kale, I. (september1996.). Decomposition of IIR transfer functions into parallel arbitrary-order IIR subfilters. IEEE Nordic Signal Processin Symposium (NORSIG'96), Dipoli Convension Center and Department of Electrical Engineering Espoo, Finland.
- [19] Shuni, Chu., Burrus, C. (Nov 1984). Multirate filter designs using comb filters. Circuits and Systems, IEEE Transactions, 31(11), 913 -924.